

UserSecurity

- [Description](#)
- [Code Sample](#)
- [Methods](#)
 - [getPropertyOptions](#)
 - [setProperties](#)
 - [getProperties](#)
 - [getDisableHashLogin](#)
 - [getAllowSessionTimeout](#)
 - [getForceSessionTimeout](#)
 - [getAuthenticateAllApi](#)
 - [encryptPassword](#)
 - [verifyPassword](#)
 - [passwordPolicies](#)
 - [validatePassword](#)
 - [validateUserOnInsert](#)
 - [validateUserOnUpdate](#)
 - [validateUserOnProfileUpdate](#)
 - [insertUserPostProcessing](#)
 - [updateUserPostProcessing](#)
 - [updateUserProfilePostProcessing](#)
 - [passwordResetPostProcessing](#)
 - [deleteUserPostProcessing](#)
 - [getLoginFormFooter](#)
 - [getUserCreationFormFooter](#)
 - [getUserEditingFormFooter](#)
 - [getUserProfileFooter](#)
 - [getUserDetailsButtons](#)
 - [loginPreProcessing](#)
 - [loginPostProcessing](#)
 - [requestPostProcessing](#)
 - [isDataExist](#)

Description

- `org.joget.directory.model.service.UserSecurity`
- Under `wflow-directory` module
- Service methods used to improve user security

Code Sample

```

import java.util.Set;
import org.joget.apps.app.service.AppUtil;
import org.joget.commons.util.SecurityUtil;
import org.joget.directory.dao.UserDao;
import org.joget.directory.dao.RoleDao;
import org.joget.directory.model.User;
import org.joget.directory.model.service.DirectoryUtil;
import org.joget.directory.model.service.UserSecurity;

UserSecurity us = DirectoryUtil.getUserSecurity();

UserDao userDao = (UserDao) AppUtil.getApplicationContext().getBean("userDao");
RoleDao roleDao = (RoleDao) AppUtil.getApplicationContext().getBean("roleDao");

//Create an User
User user = new User();
user.setId("testuser");
user.setUsername("testuser");
user.setTimeZone("0");
user.setActive(1);
user.setFirstName("Test User");
if (us != null) {
    user.setPassword(us.encryptPassword("testuser", "password"));
} else {
    user.setPassword(StringUtil.md5Base16("password"));
}
user.setConfirmPassword("password");

//set role
Set roleSet = new HashSet();
roleSet.add(roleDao.getRole("ROLE_USER"));
user.setRoles(roleSet);

userDao.addUser(user);
if (us != null) {
    us.insertUserPostProcessing(user);
}

```

Methods

getPropertyOptions

```
public java.lang.String getPropertyOptions()
```

Get JSON property options to embed into the property options of Directory Manager Plugin

setPropertyOptions

```
public void setPropertyOptions(java.util.Map<java.lang.String, java.lang.Object> propertyOptions)
```

Set property options to use in User Security Implementation

getPropertyOptions

```
public java.util.Map<java.lang.String, java.lang.Object> getPropertyOptions()
```

Get property options to use in User Security Implementation

getDisableHashLogin

```
public java.lang.Boolean getDisableHashLogin()
```

Disable Hash Login?

getAllowSessionTimeout

```
public java.lang.Boolean getAllowSessionTimeout()
```

Allow Session Timeout?

getForceSessionTimeout

```
public java.lang.Boolean getForceSessionTimeout()
```

Force Session Timeout?

getAuthenticateAllApi

```
public java.lang.Boolean getAuthenticateAllApi()
```

Authenticate all API?

encryptPassword

```
public java.lang.String encryptPassword (java.lang.String username, java.lang.String password)
```

Encrypt Password with custom methodology

verifyPassword

```
public java.lang.Boolean verifyPassword (org.joget.directory.model.User user, java.lang.String password)
```

Verify Password with custom methodology.

passwordPolicies

```
public java.util.Collection<java.lang.String> passwordPolicies ()
```

To get a set of password policies to display in user profile form.

validatePassword

```
public java.util.Collection<java.lang.String> validatePassword (java.lang.String username, java.lang.String oldPassword, java.lang.String newPassword, java.lang.String confirmPassword)
```

To validate password format and validity.

Return Collection of error message String.

validateUserOnInsert

```
public java.util.Collection<java.lang.String> validateUserOnInsert(org.joget.directory.model.User user)
```

To validate user before insert to database

Return Collection of error message String.

validateUserOnUpdate

```
public java.util.Collection<java.lang.String> validateUserOnUpdate(org.joget.directory.model.User user)
```

To validate user before updating

Return Collection of error message String.

validateUserOnProfileUpdate

```
public java.util.Collection<java.lang.String> validateUserOnProfileUpdate(org.joget.directory.model.User user)
```

To validate user before updating profile

Return Collection of error message String.

insertUserPostProcessing

```
public void insertUserPostProcessing(org.joget.directory.model.User user)
```

Processing after a user is added.

updateUserPostProcessing

```
public void updateUserPostProcessing(org.joget.directory.model.User user)
```

Processing after a user is modified.

updateUserProfilePostProcessing

```
public void updateUserProfilePostProcessing(org.joget.directory.model.User user)
```

Processing after a user profile is updated.

passwordResetPostProcessing

```
public void passwordResetPostProcessing(org.joget.directory.model.User user)
```

Processing after a user password is reset.

deleteUserPostProcessing

```
public void deleteUserPostProcessing(java.lang.String username)
```

Processing after a user is deleted.

getLoginFormFooter

```
public java.lang.String getLoginFormFooter()
```

HTML to add below login form

getUserCreationFormFooter

```
public java.lang.String getUserCreationFormFooter()
```

HTML to add below user creation form

getUserEditingFormFooter

```
public java.lang.String getUserEditingFormFooter(org.joget.directory.model.User user)
```

HTML to add below user editing form

getUserProfileFooter

```
public java.lang.String getUserProfileFooter(org.joget.directory.model.User user)
```

HTML to add below user profile form

getUserDetailsButtons

```
public java.lang.String getUserDetailsButtons(org.joget.directory.model.User user)
```

HTML to extend user details page button

loginPreProcessing

```
public void loginPreProcessing(org.joget.directory.model.User user, java.lang.String password)
```

Processing before user login authentication

loginPostProcessing

```
public void loginPostProcessing(org.joget.directory.model.User user, java.lang.String password, java.lang.Boolean loggedIn)
```

Processing after user login authentication.

requestPostProcessing

```
public void requestPostProcessing()
```

Processing after HTTP Request Completed.

isDataExist

```
public boolean isDataExist(java.lang.String username)
```

Check whether a custom security data is exist for a user.