

General login service

Summary

Login is largely classified into two methods. The first one is the login through ID and password prepared at the time of subscription (to be called general login hereafter) and the second one is the login through the certificate such as NPKI or GPKI (to be called GPKI login hereafter). For general login, the screen is constructed differently in accordance with the classification of user (general member, corporate member and business user).

- Flow of functions

Name of function	Flow of function
General login	ID/password input → Login request → Authority inquiry → Session configuration → Login log creation → Menu configuration for each authority → Screen loading for each authority

- Security items when using login component

Provided login component does not furnish section encryption (encryption between browser and server). (Section encryption cannot be provided because this is the server side component.) For this, maintaining the security of login information is recommended with the method of introducing separate section encryption solution or applying SSL to login request when implementing with common component.

Description

This is the service component to allow the access of the user and to support the securing of information for the user who attempts to connect to the system or application program for the purpose of security, and it is the method of processing the certification of the user by inquiring the already saved user information with the received login information of the logged in user (user ID/password).

Package Dependency

Login Package has direct functional dependency only for the common package (cmm) of element technology. However, in order to be executed without error during the component distribution, the distribution file is constituted with format/date/calculation, mail connection interface, and system package in accordance with the dependency between packages.

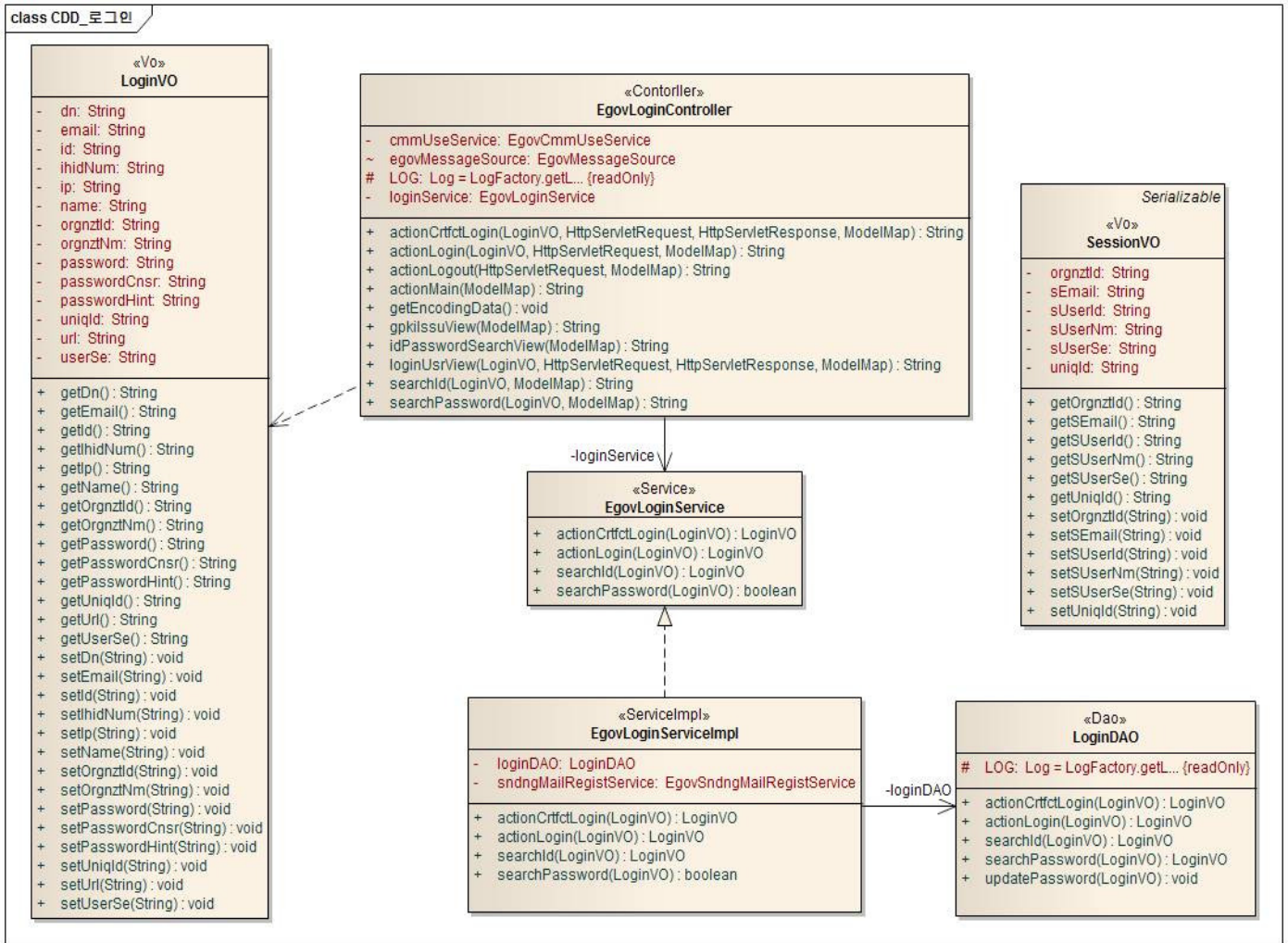
- Dependency between packages: [Package Dependency on user directory/integrated certification](#)

Related Sources

Type	Name of Corresponded Source	Remarks
Controller	egovframework.com.uat.uia.web.EgovLoginController.java	Controller class processing general login, certificate login
Service	egovframework.com.uat.uia.service.EgovLoginService.java	Business interface class processing general login, certificate login
ServiceIm	egovframework.com.uat.uia.service.impl.EgovLoginServiceImpl.java	Business

pl		implementation class processing general login, certificate login
VO	egovframework.com.cmm.SessionVO.java	VO class for session
VO	egovframework.com.cmm.LoginVO.java	VO class for login
DAO	egovframework.com.uat.uia.service.impl.LoginDAO.java	DAO class processing general login, certificate login
JSP	WEB_INF/jsp/egovframework/com/uat/uia/EgovLoginUsr.jsp	Login certification JSP page
JSP	WEB_INF/jsp/egovframework/com/uat/uia/EgovIdPasswordSearch.jsp	JSP page for ID/password searching
JSP	WEB_INF/jsp/egovframework/com/uat/uia/EgovIdPasswordResult.jsp	JSP page for ID/password searching result
Query XML	resources/egovframework/sqlmap/com/uat/uia/EgovLoginUsr_SQL_Mysql.xml	QUERY XML for MySQL for general login, certificate login
Query XML	resources/egovframework/sqlmap/com/uat/uia/EgovLoginUsr_SQL_Oracle.xml	QUERY XML for Oracle for general login, certificate login
Query XML	resources/egovframework/sqlmap/com/uat/uia/EgovLoginUsr_SQL_Tibero.xml	QUERY XML for Tibero for general login, certificate login
Query XML	resources/egovframework/sqlmap/com/uat/uia/EgovLoginUsr_SQL_Altibase.xml	QUERY XML for Altibase for general login, certificate login
Message properties	resources/egovframework/message/com/message-common_ko_KR.properties	Message properties for general login, certificate login

Class Diagram



Related Table

Name of table	Name of table (English)	Remarks
Business user	COMTNEPLYRINFO	Manage business user information.
Corporate member	COMTNENTRPRSMBER	Manage corporate user information. Applicant's inhabitant identification number (APPLCNT_IHIDNUM) column is existed, however, it is excluded from data saving and inquiry logic, therefore, it shall be implemented by encryption if required.
General member	COMTNGNRLMBER	Manage general user information. inhabitant identification number (APPLCNT_IHIDNUM) column is existed, however, it is excluded from data saving and inquiry logic, therefore, it shall be implemented by encryption if required.

Configuration

For implementation method of general login function, there are two methods, i.e., the certification method with session and the certification with spring security framework. Configuration for this can be changed through revision of context-egovuserdetailshelper.xml and web.xml files, and the configuration for the page to be transferred can be revised through the configuration of globals.properties file.

Transfer page configuration after login success

When login is successfully performed, the page to be transferred can be configured through the `globals.MainPage` property value of `globals.properties` file. It can be configured as follows, and if login is succeeded by the user, it is moved to the `/EgovContent.do` page.

```
# MainPage Setting
Globals.MainPage = /EgovContent.do
```

Session method configuration

1) Revision of `context-egovuserdetailshelper.xml` : After releasing the note of `egovUserDetailsSessionService` bean configuration as follows, register the `egovUserDetailsSessionService` bean to the `egovUserDetailsService` property of the `egovUserDetailsHelper` bean.

```
<!-- Service class with which the LoginVO, authority, and the state of certification of certified user
can be confirmed -->
<bean id="egovUserDetailsHelper" class="egovframework.com.cmm.util.EgovUserDetailsHelper">
  <property name="egovUserDetailsService">
    <!-- Apply the ref tag to the underside of the desired bean id. -->
    <ref bean="egovUserDetailsSessionService" />
  </property>
</bean>

<!-- 0. The bean which uses the certification (dummy) for the test except session/security -->
<!--<bean id="egovTestUserDetailsService"
class="egovframework.com.cmm.service.impl.EgovTestUserDetailsServiceImpl"/>-->

<!-- 1. The bean which uses the certification with session -->
<bean id="egovUserDetailsSessionService"
class="egovframework.com.cmm.service.impl.EgovUserDetailsSessionServiceImpl"/>

<!-- 2. The bean which will use the certification with security -->
<!--<bean id="egovUserDetailsService"
class="egovframework.com.sec.ram.service.impl.EgovUserDetailsServiceImpl"/>-->
```

For session method, access control can be conducted at the controller by adding the code as follows.

```
Boolean isAuthenticated = EgovUserDetailsHelper.isAuthenticated();
if(!isAuthenticated) {
  model.addAttribute("message", egovMessageSource.getMessage("fail.common.login"));
  return "egovframework/com/uat/uia/EgovLoginUsr";
}
```

2) Revision of `web.xml`: This can be used based on the default configuration without any revision. (The spring security related configuration as follows is in the state of furnishing with notes.)

Configuration with spring security framework

The spring security related configuration shall always be included. The `context-security.xml` file shall be located under the `src/main/resources/egovframework/spring/com` folder.

1) Revision of `context-egovuserdetailshelper.xml`: After releasing the notes of `egovUserDetailsService` bean configuration as follows, register the `egovUserDetailsService` bean to the `egovUserDetailsService` property of the `egovUserDetailsHelper` bean.

```
<!-- The service class with which LoginVO, authority, and the state of certification of certified user can
be confirmed -->
<bean id="egovUserDetailsHelper" class="egovframework.com.cmm.util.EgovUserDetailsHelper">
  <property name="egovUserDetailsService">
    <!-- Apply the desired bean id to the following ref tag. -->
```

```

        <ref bean="egovUserDetailsSecurityService" />
    </property>
</bean>

<!-- 0. The bean which uses certification (dummy) for the test except session/security -->
<!--<bean id="egovTestUserDetailsService"
class="egovframework.com.cmm.service.impl.EgovTestUserDetailsServiceImpl"/>-->

<!-- 1. The bean which uses the certification with session-->
<!--<bean id="egovUserDetailsService"
class="egovframework.com.cmm.service.impl.EgovUserDetailsServiceImpl"/>-->

<!-- 2. The bean which will use the certification with security -->
<bean id="egovUserDetailsSecurityService"
class="egovframework.com.sec.ram.service.impl.EgovUserDetailsSecurityServiceImpl"/>

```

2) Revision of web.xml: Release the notes at the following configuration session. In the case of the EgovSpringSecurityLoginFilter filter, if login certification is failed, the screen which is returned to the user is being inputted to the parameter. If login to loginURL parameter value is failed, input the url route to be returned.

Certification method change and extension

If certification method is to be used by applying the method other than the session and spring security, it can be used by implementing the EgovUserDetailsService interface.

```
package egovframework.com.cmm.service;
```

```
public interface EgovUserDetailsService {
```

```

    /**
     * Fetch the certified user object with VO form.
     * @return Object -User ValueObject
     */
    public Object getAuthenticatedUser();

    /**
     * Fetch the authority information of certified user.
     * Ex) [ROLE_ADMIN, ROLE_USER, ROLE_A, ROLE_B, ROLE_RESTRICTED,
    IS_AUTHENTICATED_FULLY, IS_AUTHENTICATED_REMEMBERED,
    IS_AUTHENTICATED_ANONYMOUSLY]
     * @return List - User authority information list
     */
    public List<String> getAuthorities();

    /**
     * Check whether it is the certified user or not.
     * @return Boolean - State of the certified user (TRUE / FALSE)
     */
    public Boolean isAuthenticated();

```

```
}
```

In order to apply EgovUserDetailsService implementation class, conduct the mapping of EgovUserDetailsService implementation class to egovUserDetailsService property value when configuring egovUserDetailsHelper bean after registering the EgovUserDetailsService implementation class to the bean in the first place.

Following configuration is the example of registering as the property of egovUserDetailsHelper bean which is used for the user certification after registering dummy certification class file to the bean.

```

<!-- The service class with which LoginVO, authority, and the state of certification of certified user
can be confirmed -->
<bean id="egovUserDetailsHelper" class="egovframework.com.cmm.util.EgovUserDetailsHelper">
    <property name="egovUserDetailsService">
        <!-- Apply the desired bean id to the following ref tag. -->
        <ref bean="egovTestUserDetailsService" />
    </property>
</bean>

<!-- 0. The bean which uses the certification (dummy) for the test except session/security -->
<bean id="egovTestUserDetailsService"
class="egovframework.com.cmm.service.impl.EgovTestUserDetailsServiceImpl"/>

```

Example of certification implementation with session

The `getAuthenticatedUser()` method returns the VO object which possessed certified user information. In the common component, all of the user information is utilized by applying the `egovframework.com.cmm.LoginVO` class, therefore, implement the method so that the `egovframework.com.cmm.LoginVO` object is to be returned. Following is the example of obtaining and returning the loginVO object from the session by utilizing the spring framework.

```

public Object getAuthenticatedUser() {

    return RequestContextHolder.getRequestAttributes().getAttribute("loginVO",
RequestAttributes.SCOPE_SESSION);

}

```

The `getAuthorities()` method returns the authority possessed by the user by putting in the list object. The authority information which can be used at the authority management package is returned by putting in the list, and in the following example, the authority information is exceptionally returned by containing them arbitrarily in the list.

```

public List<String> getAuthorities() {

    List<String> listAuth = new ArrayList<String>();
    listAuth.add("IS_AUTHENTICATED_ANONYMOUSLY");
    listAuth.add("IS_AUTHENTICATED_FULLY");
    listAuth.add("IS_AUTHENTICATED_REMEMBERED");
    listAuth.add("ROLE_ADMIN");
    listAuth.add("ROLE_ANONYMOUS");
    listAuth.add("ROLE_RESTRICTED");
    listAuth.add("ROLE_USER");

    return listAuth;

}

```

The `isAuthenticated()` method is used to check whether current user is the certified user or not. In the following example, the state of certification is inspected through the checking whether there is the loginVO value in the session.

```

public Boolean isAuthenticated() {

    if (RequestContextHolder.getRequestAttributes() == null) {
        return false;

    } else {

        if (RequestContextHolder.getRequestAttributes().getAttribute(
            "loginVO", RequestAttributes.SCOPE_SESSION) == null) {
            return false;
        }
    }
}

```

```

    } else {
        return true;
    }
}

```

```

}

```

* Reference: The egovframework.com.cmm.service.impl.EgovUserDetailsSessionServiceImpl class of common component which implements certification service with session

Related Functions

User general login

Business Rule

General login inquires user information for each user classification with ID and password. Login log is created through the inquired user information.

Related Code

N/A

Screen and execution manual

Action	URL	Controller method	QueryID
Login screen	/uat/uia/egovLoginUsr.do	loginUsrView	
General login	/uat/uia/actionLogin.do	actionLogin	"loginDAO.ssoLoginByEsntId"

▶ 일반로그인



- Selection of business classification: Select user business classification.
- ID input: Input the ID.
- Password input: Input the password.
- Login: Perform the user login certification work through ID and password.
- Member subscription: Move to the member subscription page which suits to the selected business classification.
- ID/password finding: Move to the ID/password finding page.
- Certificate login: Call the GPKI certificate login pop-up.
- Certificate guide: Call the GPKI certificate guide pop-up.
- ID saving: Remember ID information at the next connection.