Home > WebSphere Application Server Liberty base > Securing Liberty and its applications > Developing extensions to the Liberty security infrastructure >

Previous Next

Developing a programmatic login for obtaining authentication data

Search in all products

Search in this product...



≡ Table of Contents Change version or product ~

> ₱ Print ■ PDF ∨ ② Help Take a tour

You can use the Java Authentication and Authorization Service (JAAS) login framework to obtain the authentication data from your application.

About this task

Your application can perform a JAAS programmatic login using the DefaultPrincipalMapping JAAS context entry name to obtain a Subject object with a javax.resource.spi.security.PasswordCredential instance in the private credentials set that contains the user name and password configured for an authData element.

Procedure

1. Add the appSecurity-2.0 and passwordUtilities-1.0 features in the server.xml file. For example:



```
<featureManager>
   <feature>appSecurity-2.0</feature>
   <feature>passwordUtilities-1.0</feature>
</featureManager>
```

2. Configure an authData element in the server.xml file. For example:



```
<authData id="myAuthData" user="myUser" password="myPassword"/> <!</pre>
Encode the password within the configuration. You can get the encode
```

3. Perform a programmatic login with the DefaultPrincipalMapping JAAS login context entry name from your application servlet or enterprise bean, replacing the mapping alias with the one you need. For example:

```
HashMap map = new HashMap();
map.put(com.ibm.wsspi.security.auth.callback.Constants.MAPPING_ALIA
CallbackHandler callbackHandler = new com.ibm.wsspi.security.auth.
LoginContext loginContext = new LoginContext("DefaultPrincipalMapp:
loginContext.login();
Subject subject = loginContext.getSubject();
Set<javax.resource.spi.security.PasswordCredential> creds = subjec
PasswordCredential passwordCredential = creds.iterator().next();
```

Note

The error handling is not shown for simplicity. A javax.security.auth.login.LoginException is returned if the authentication alias requested does not exist or is malformed.

4. Obtain the user name and password from the PasswordCredential. For example:

```
String userName = passwordCredential.getUserName();
char[] password = passwordCredential.getPassword();
// Do something with the userName and password.
```



5. If Java 2 Security is enabled, then the application must be granted the javax.security.auth.PrivateCredentialPermission.For example, grant the permission in the application's META-INF/permissions.xml file to access the PasswordCredential object:

```
<?xml version="1.0" encoding="UTF-8"?>
<permissions xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi=</pre>
  <permission>
    <class-name>javax.security.auth.PrivateCredentialPermission/cl
    <name>javax.resource.spi.security.PasswordCredential * "*"</nar</pre>
    <actions>read</actions>
  </permission>
```

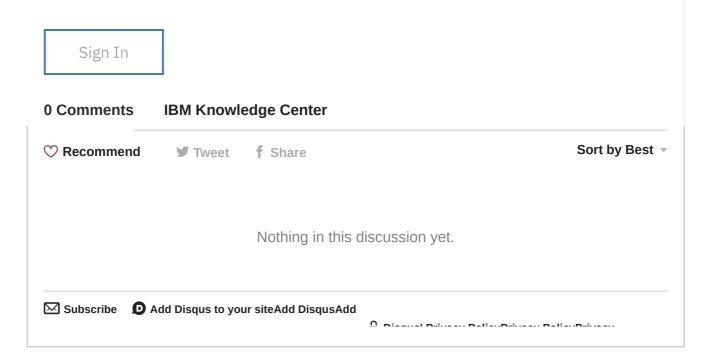
```
<!-- Other permissions -->
</permissions>
```

For more information about Java 2 Security, see Liberty: Java 2 Security.

Parent topic:

→ Developing extensions to the Liberty security infrastructure

Please note that DISQUS operates this forum. When you sign in to comment, IBM will provide your email, first name and last name to DISQUS. That information, along with your comments, will be governed by DISQUS' privacy policy. By commenting, you are accepting the DISQUS terms of service.



Related topics:

Developing a custom TAI for Liberty

Customizing user registries or repositories for Liberty

Developing JAAS custom login modules for a system login configuration

Developing a custom JASPIC authentication provider for Liberty

Developing a Java Authorization Contract for Containers (JACC) Authorization Provider

Developing a customPasswordEncryption Provider

Do you want to...

Open a ticket and download fixes at the IBM Support Portal

Find a technical tutorial in **IBM Developer**

Find a best practice for integrating technologies in IBM Redbooks

Explore, learn and succeed with training on the IBM Skills Gateway

Privacy Terms of use Accessibility Feedback Cookie preferences Contact

