icw
eHEALTH FRAMEWORK

eHealth Framework

How to create your own module

Notice

The wording in this document applies equally to women and men. The masculine form was selected to ease the comprehensibility and legibility of the text.

All company logos are a registered trademark of InterComponentWare AG.

The product names mentioned in this documentation are either trademarks or registered trademarks of the respective owners and are stated for identification purposes only.

Note:

The current version of this document has a draft status and various chapters are still in review.

The document is collaboratively built with the use of the Darwin-Information-Typing-Architecture (DITA) and has therefore a draft status concerning styles and layout. The necessary adaptations are currently also in a developmental stage.

Contents

1 Overview

Purpose

This how to describes how to enable and use a LDAP integration in an eHF-based assembly.

Scope

The following picture shows the scope of this how to.

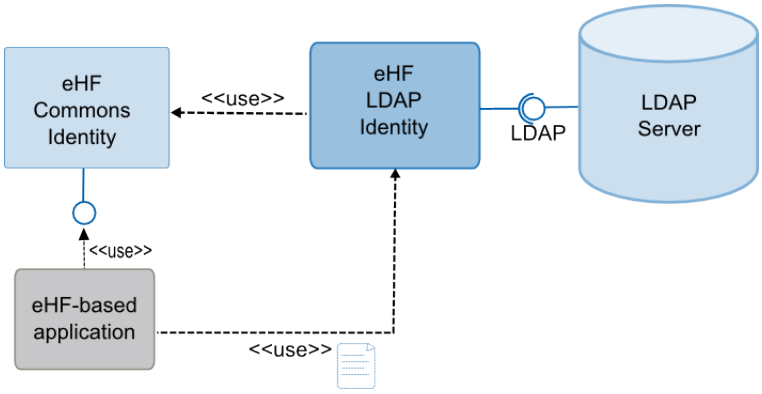

Figure 1: Scope

For more details on the eHealth Framework please refer to the eHealth Framework Reference Documentation.

2 Enabling and using the eHF LDAP integration in an eHF-based assembly

2.1 Insert needed dependencies

Preparation of the needed dependencies in the assembly

1. Insert needed eHF-based dependencies in the eHF-based assembly

   Add the following eHF module dependencies (here: SNAPSHOT) to the `project.xml` of your eHF assembly:

```
<dependency>
            <groupId>ehf</groupId>
            <artifactId>ehf-commons-identity-runtime</artifactId>
            <version>SNAPSHOT</version>
            <properties>
                <war.bundle>true</war.bundle>
            </properties>
            <type>jar</type>
        </dependency>
        <dependency>
            <groupId>ehf</groupId>
            <artifactId>ehf-ldap-identity-config</artifactId>
            <version>SNAPSHOT</version>
            <properties>
                <ehf-module>ldap-identity</ehf-module>
            </properties>
            <type>jar</type>
        </dependency>
        <dependency>
            <groupId>ehf</groupId>
            <artifactId>ehf-ldap-identity-runtime</artifactId>
            <version>SNAPSHOT</version>
            <properties>
                <war.bundle>true</war.bundle>
            </properties>
            <type>jar</type>
        </dependency>
```

   The eHF-based assembly contains the needed dependencies to the needed eHF modules.

2. Insert the needed third-party dependencies in the eHF-based assembly

   Since the implementation of the eHF LDAP integration bases on the Spring LDAP project (here: V. 1.3.0.RELEASE), you have also to add the following dependency to your project::

   Open the `project.xml` file of the eHF-based assembly and insert following code snippet:

```
 <dependency>
            <groupId>org.springframework.ldap</groupId>
            <artifactId>spring-ldap-core</artifactId>
            <version>1.3.0.RELEASE</version>
            <type>jar</type>
            <properties>
                <war.bundle>true</war.bundle>
            </properties>
        </dependency>
        <!-- Spring Ldap third party lib dependencies -->
```

```
        <dependency>
            <groupId>com.sun</groupId>
            <artifactId>ldapbp</artifactId>
            <version>1.0</version>
            <type>jar</type>
            <properties>
               <war.bundle>true</war.bundle>
            </properties>
        </dependency>
```

The eHF-based assembly contains the needed third-party dependencies.

2.2 Tomcat configuration

Tomcat Configuration when authentication against LDAP system is needed

1. Basic configuration

   If you want to authenticate your application users against the LDAP system, you have to add the following code fragment to your JAAS login configuration file (for **Tomcat** : `conf/catalina.login`):

```
ehf {

    // Login module
    com.icw.ehf.authentication.jaas.DefaultLoginModule required

    // Login module options
    logoutHandler.1 = com.icw.ehf.authorization.service.
PermissionLogoutHandler
    adapter=com.icw.ehf.ldap.identity.service.AuthenticationService
    userServiceName=ldapUserService
    providerUrl="ldap://localhost:50000/dc=bundestag,dc=de";

};
```

   The **adapter** parameter specifies the implementation class of AuthenticationAdapter interface that performs the authentication against an LDAP system. This class is provided by the eHF LDAP Identity module. The **userServiceName** parameter is the Spring bean name of the LDAP implementation of the `UserService` interface. This bean is exposed by the eHF LDAP Identity module. The **providerUrl** parameter specifies the URL of your target LDAP system (in this example: `ldap://localhost:50000/dc=bundestag,dc=de`)

   The authentication against a LDAP system is configured.

2. LDAP server specific configuration

   Some LDAP server (like ApacheDS) support only dn-based user names (like `uid=jackerma,ou=personen,dc=bundestag,dc=de`), so if you want to support user-friendly user name input, you can use two parameter **userNamePrefix** and **userNamePostfix** .

```
 ehf {

    // Login module
    com.icw.ehf.authentication.jaas.DefaultLoginModule required

   // Login module options
   logoutHandler.1 = com.icw.ehf.authorization.service.
PermissionLogoutHandler
   adapter=com.icw.ehf.ldap.identity.service.AuthenticationService
   userNamePrefix="uid="
   userNamePostfix=",ou=personen,dc=bundestag,dc=de"
```

```
   userServiceName=ldapUserService
   providerUrl="ldap://localhost:50000/dc=bundestag,dc=de";

};
```

The configuration for the authentication against a LDAP system contains LDAP server specific properties.

3. Extended configuration to support more than one identity provider system

   If you want to support more than one identity provider system with your application, for example a LDAP-based identity provider and the eHF Usermgnt user database, you can do this by specifying parameters for additional adapters as shown in the following configuration snippet:

```
ehf {

   // Login module
   com.icw.ehf.authentication.jaas.DefaultLoginModule required

       // Login module options
       logoutHandler.1 = com.icw.ehf.authorization.service.
PermissionLogoutHandler
       adapter=com.icw.ehf.ldap.identity.service.AuthenticationService
       userServiceName=ldapUserService
       providerUrl="ldap://localhost:50000/dc=bundestag,dc=de"
       additionalAdapter.1=com.icw.ehf.usermgnt.service.
AuthenticationService
       additionalAdapter.1.userServiceName=usermgntUserService;
};
```

The **additionalAdapter.1** parameter is analog to the adapter parameter and specifies the implementation class of the `AuthenticationAdapter`. All other parameters that are prefixed with **additonalAdapter.1.** belong to this adapter. You can specify as many additional identity providers as you need to support in your application. The users are authenticated against the specified identity providers in the order in which they are listed in the JAAS configuration file here.

The configuration for the authentication contains properties for the support of more than one identity provider system.

2.3 LDAP Connection Configuration

LDAP connection configuration

1. Basic configuration

   Queries against the LDAP system are executed in the context of a technical user. The connection data for this technical user must be specified on assembly-level within the `configuration/configuration.deployment.properties` file as shown in this configuration snippet:

```
# ldap connection settings
connection.ldap.url=ldap://localhost:50000
connection.ldap.base=dc=bundestag,dc=de
connection.ldap.user.dn=jackerma@bundestag.de
connection.ldap.password=password
```

   The basic connection configuration to a LDAP system is done.

2. LDAP server specific configuration

   In `src/main/config/ehf-system-context.xml` you must configure the bean `ldapTemplate`.

```
 <bean id="ldapTemplate" class="org.springframework.ldap.core.LdapTemplate">
        <property name="contextSource">
            <bean class="org.springframework.ldap.core.support.
LdapContextSource">
                <property name="url" value="@@connection.ldap.url@@" />
                <property name="base" value="@@connection.ldap.base@@" />
                <property name="userDn" value="@@connection.ldap.user.dn@@" /
>
                <property name="password" value="@@connection.ldap.
password@@" />
                @@@ldap.base.enviroment.properties.apacheds.fragment@@@
            </bean>
        </property>
    </bean>
```

The module **ldap-identity** currently provides two fragments `ldap.base.enviroment.properties.apacheds.fragment` (ApacheDs-specific configuration) and `ldap.base.enviroment.properties.adam.fragment` (ADAM-specific configuration). If you use another LDAP-Server, then replace the fragement (in the example `@@@ldap.base.enviroment.properties.apacheds.fragment@@@`) by your configuration.

The LDAP server specific configuration is done.

2.4 Tomcat configuration

Enabling SSL Configuration if it is needed

1. SSL configuration in eHF assembly

   Configure using the SSL LDAP server endpoint (ldaps:) e.g. `providerUrl="ldaps://localhost:50000/dc=bundestag,dc=de"` and `connection.ldap.url=ldaps://localhost:50000`

   The SSL configuration in eHF assembly is ready.

2. SSL configuration in tomcat

   Configure the global keystore to accept the LDAP server SSL certificate e.g. `java -Djavax.net.ssl.trustStore=./src/test/resources/LdapClientTruststore.jks -Djavax.net.ssl.trustStorePassword=changeit ...`. In ehf deployment this means adding this certificate to the already present Tomcat truststore: `keytool -importcert -trustcacerts -alias ldapserver1 -file ldapserver.crt -keystore tomcat.truststore` Furthermore, please check whether the following enviroment variable is set correctly: `CATALINA_OPTS="-Djavax.net.ssl.trustStore=your_path_to/truststore.jks -Djavax.net.ssl.trustStorePassword=your_password"` This is necessary for Tomcat to know in which trust store it must be looked.

   The SSL configuration in tomcat.

2.5 LDAP Identity Configuration

LDAP Identity configuration

1. LDAP Identity configuration

   The module **eHF LDAP Identity** owns a default module configuration (see below) which is ADAM specific.

```
#ADAM attribute configuration
```

```
user.guid=objectGUID
user.domain=objectGUID
user.identifier.username=userPrincipalName
user.identifier.guid=objectGUID
user.identifier.domain=objectGUID

userGroup.guid=objectGUID
userGroup.name=cn

person.guid=objectGUID
person.domain=objectGUID
person.firstName=givenName
person.lastName=sn
person.title=title
person.telephone.value=telephoneNumber
person.email.value=mail

#System configuration
userGroup.base.dn=
system.domain=system
userGroup.objectclass=group
object.userGroup.name=Roles
member.attribute=memberOf
ldap.server.type=ADAM
```

If you want to overwrite this configuration, you have to insert a property file named `ldap-identity.properties` in the assembly under `scr/main/resources/META-INF` and fill it with the properties which should be overwritten.

The LDAP Identity module configuration is done.