Lab: Integration Point Authentication

In this lab, you will replace the base application's user authentication integration with a plugin and associated class that validated authentication information from an external system. The external system in this lab is ExternalApp. ExternalApp will function as an authentication system similar to an LDAP system.

Requirements

This lab requires that you use TrainingApp 8.0, ExternalApp 8.0, Guidewire Studio 8.0, and a supported web browser. Start Guidewire Studio for Training App. Debug the Server.

To run ExternalApp, navigate to c:\Guidewire\ExternalApp and double-click the Start ExternalApp shortcut. The command window of ExternalApp will read "\*\*\*\* ContactManager ready \*\*\*\*" when the application is running.

To view, edit, and delete various contacts, log in to TrainingApp as Super User. The default URL for TrainingApp is <http://localhost:8880/ab/ContactManager.do>. The login/password for Super User is su/gw.

1. Create the authentication service plugin

In this exercise, you will write a Gosu class that implements gw.plugin.security.AuthenticationServicePlugin and create an authentication service plugin registry for your class.

In your Gosu class you will override the authenticate() method to take a UserNamePasswordAuthenticationSource that contains the http request's username and password parameters (key value pair). You will then supply these value to an ExternalApp's User Authentication web service. The web service contains one method — authenticate(username, password) — that returns a Boolean value of true when the values match. Your authenticate() method will return the user's publicID for when the web service method returns true.

ExternalApp's User Authentication web service authenticates the following key value pairs:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User Name | Password |  | User Name | Password |
| aapplegate | guidewire |  | elee | guidewire |
| bbaker | guidewire |  | mmaples | guidewire |
| cclark | guidewire |  | ssmith | guidewire |
| ccraft | guidewire |  | su | guidewire |
| dhenson | guidewire |  |  |  |

Tasks

1. Create a package named acmelab.ta.plugin.authentication
2. Create the package in the correct plugin directory.
3. In the package, create a web service connection
4. Create a web service collection that connects to ExternalApp's User web service.
5. Use the following WSDL:  
   <http://localhost:9990/ab/ws/external/UserAuthenticationAPI?WSDL>
6. Create an implementation of the authentication service class that meets the following requirements:
7. Create your plugin class in the authentication package.
8. Employ Guidewire recommended naming conventions for your plugin class.
9. The authenticate() method must authenticate the user based on the return value of the web service authenticate() method.
10. The credentials for the web service must have soap admin user permissions.
11. The setCallback method simply sets the callback handler.
12. If authentication is successful, returns the public ID for the user in the authenticate() method.
13. If unsuccessful, throws a javax.security.auth.login.FailedLoginException in the authenticate() method.
14. Create the AuthenticationServicePlugin registry file
15. In the Plugins registry, create a new plugin named AuthenticationServicePlugin.
16. Specify the AuthenticationServicePlugin interface.
17. Add and specify your Gosu Plugin class.
18. Deploy your changes
19. Restart the server from Studio by stopping a running server and then running the server.

Verification

Verification requires that you run ExternalApp. You will now verify that your plugin meets the requirements.

1. Login into TrainingApp using the username/ password key value of aapplegate/guidewire
2. Confirm that successful login
3. Login into TrainingApp using the username/ password key value of su/guidewire
4. Confirm that successful login
5. Login into TrainingApp using the username/ password key value of aapplegate/gw
6. Confirm that failed login
7. Login into TrainingApp using the username/ password key value of aFakeName/guidewire
8. Confirm that failed login

Restoring TrainingApp

After verification of your exercise, disable the Authentication Service Plugin registry file. Then restart the application server by stopping the running server and then running the server in either Run or Debug mode.

If you do not disable the plugin, then you will need to have ExternalApp running any time that you wish to log on to TrainingApp, and you'll have to use the password "guidewire" instead of "gw".

|  |  |
| --- | --- |
|  | Stop and ask your instructor to review your completed lab. |