

Table of Contents

Setting Up Oauth Protection 2

Callers' Workflow 2

Setting Up Oauth Protection

If your application exposes at least one [service](#) (REST-ful or SOAP) that needs to be protected by the OAuth2.0 protocol, you need to deploy a special web application that will manage customers and issue access and refresh tokens. To deploy such an application, you will need to do the following:

1. Go to the Edit menu of the Configuration Tool
2. Select the entry with the name "SERVICE PROTECTION"

Note that the deployment of the dynamic feature should only be done once.

Callers' Workflow

This section describes how a third party (client) should call an Aware IM service protected by OAuth.

1. Before the client can call protected services, he must register himself with the OAuth server (this is done only once). To register, the client has to call this URL:

```
http(s)://ServerName:ServerPort/CXF/forms/registerClient.jsp
```

This should display a form where a client has to enter some mandatory and some optional values. The only value that requires explanation is "redirect URI". This defines the URL that the OAuth server should call after creating an authorization token and returning to the client's application, for example

```
http://ClientServerName/SomePath
```

1. When the form is submitted a reply is returned with the unique client id and secret. Client should store these values and use them when calling services
2. When calling a service to start a new authentication process the client has to go to this URL:

```
http(s)://ServerName:ServerPort/CXF/services/oauth/authorize?client_id=...&response_type=code&redirect_uri=...&scope=...
```

(redirect_uri is optional – if not specified, values registered for the client will be used) (if a scope is a general scope, provide scope as BusinessSpaceName_general)

1. The above URL call should return an "authorization code by calling the "redirect_uri" and supplying it with the "code" parameter. This code can then be exchanged for access token. To get access token the client has to call this URL using POST and supply the received code:

```
http(s)://ServerName:ServerPort/CXF/services/oauth/token?grant_type=authorization_code&code=...&client_id=...&client_secret=...&redirect_uri=...
```

1. HTTP headers of the request should be:

1. Accept: application/json"

2. Content-Type: application/x-www-form-urlencoded

1. Access token is returned as JSON string that looks like this:

```
{ "tokenKey": "...", "tokenType": "Bearer", "refreshToken": "...", "expiresIn": 3600, "issuedAt": -1, "issuer": null, "encendedToken": null, "parameters": {}, "approvedScope": "general" }
```

1. The client should extract tokenKey and refreshToken (if he wants to refresh tokens). The client can then call Aware IM service with the provided token like so:

```
http://ServerName:ServerPort/CXF/services/serviceCall/api?serviceParameters
```

1. The following HTTP headers must be provided with the call:

1. Authorization=[Bearer accessToken]
2. AW_DOMAIN=Name of business space
3. AW_SERVICE=Name of Aware IM service

REFRESH TOKENS: Refresh tokens are returned along with access token in a JSON string as part of the access token request:

```
{ "tokenKey": "...", "tokenType": "Bearer", "refreshToken": "...", "expiresIn": 3600, "issuedAt": -1, "issuer": null, "encendedToken": null, "parameters": {
```

A refresh token can then be exchanged for a new access token once the original access token expires with the following call (refresh token is specified in the refresh_token parameter):

```
http://ServerName:ServerPort/CXF/services/oauth/token?grant_type=refresh_token&refresh_token=...&client_id=...&client_secret=...&redirect_uri=...
```

Same reply as for the access token request should be received with the new access token and new refresh token.

SCOPES: 1. Client must specify scope(s) in the authorization request. Requested scopes are separated by a space symbol (&scope=scope1%20scope2) 2. All requested scopes must be registered for the client. If no scopes are requested in the call, registered scopes are used. 3. The category of the called service must match one the approved scopes (approved by the end user) 4. Example of a valid service category: BSPrefix_scope1_scope2_scope3 (provided that scope3 is not "general")

CLIENT CREDENTIALS FLOW: The above describes "authorization" work flow. To use "client credentials workflow" do the following: In this flow you just need to call the AccessTokenService with a particular grant type. So you don't need to call the authorization service as the first step. The first step should be calling AccessTokenService but instead of the authorization_code grant type you specify client_credentials grant type: instead of this:

```
http://ServerName:ServerPort/CXF/services/oauth/token?grant_type=authorization_code&code=...&client_id=...&client_secret=...&redirect_uri=...
```

You specify this:

```
http://ServerName:ServerPort/CXF/services/oauth/token?grant_type=client_credentials&client_id=...&client_secret=...
```

For this to work a special ClientCredentialsGrantHandler needs to be registered. Again in theory this can be done by modifying the file AwareIM/Tomcat/webapps/CXF/WEB-INF/awareim.xml. You need to add the following:

```
<bean id="clientCredentialsGrantHandler"
class="com.awaresoft.oauthcxf.services.AwareIMCCGrantHandler">
    <property name="dataProvider" ref="oauthProvider"/>
</bean>
```

And then modify the following lines to add reference to the bean above (line in bold below):

```
<bean id="accessTokenService"
class="com.awaresoft.oauthxf.services.AwareIMAccessTokenService">
  <property name="dataProvider" ref="oauthProvider"/>
  <property name="grantHandlers">
    <list>
      <ref bean="refreshGrantHandler"/>
      <ref bean=" clientCredentialsGrantHandler  "/>
    </list>
  </property>
</bean>
```

The rest is the same as for the Authorization flow. So the client registers as before and gets the client id and secret. Then he calls access token service as explained above with the same headers as for the Authorization flow. He will then get the access token in the same way as in the authorization flow (there is no refresh token!) Then the client can call an Aware IM service using this access token in the same way it is done in the Authorization flow.

Your service documentation needs to convey to the callers of the service that before calling the service, they need to do the following:

1. Obtain an access token
2. include the access token in the Authorization Bearer HTTP header, for example (

```
Authorization: Bearer <access token>
```

).

3. The name of the business space must be in the HTTP header called AW-DOMAIN, for example

```
AW-DOMAIN: CRM
```

4. The name of the service to call must be in the header called AW-SERVICE for a REST-ful service (for example,

```
AW-SERVICE: AddCustomer
```

) and AW-SERVICE-SOAP for a SOAP service - for example

```
AW-SERVICE-SOAP: AddCustomer
```

.

From:
<http://www.awareim.com/dokuwiki/> - **Documentation**

Permanent link:
http://www.awareim.com/dokuwiki/docs/2500_config_apps/setting_up_oauth_protection?rev=1783411241

Last update: **2026/07/07 08:00**

