

2.1.4 - Association Anywhere Authentication Web Service

ACGI has developed an authentication web service that allows permitted third-party sites to perform user authentication through the Association Anywhere database. This web service does require some back-end engineering set up, but does not require an ODBC connection.

Web service Details

Authentication web service with internal IP white list validation of the referring site(s) can be referenced as http or https. ACGI recommends implementing this with SSL to avoid risk of transferring plain text or cost and complexity of encryption.

See APPENDIX A for descriptions and characteristics of data elements referenced within this document.

See APPENDIX B for Error Conditions and Error Codes

***censsawebsvclib.authentication* passing arguments as an xml document with name "p_input_xml_doc"**

Input document to validate by username and password, cust id and last name, or alias and last name (methods must be enabled via system parameters to be used):

```
<?xml version="1.0"?>
<authentication-request>
  <cust-id></cust-id>
  <last-nm></last-nm>
  <alias></alias>
  <username></username>
  <password></password>
  <session-id></session-id>
</authentication-request>
```



alias & cust-id and last-nm & cust-id authentication methods supported as of AA 6.5. Alias type used for authentication must be defined in system parameter SSA_AUTH_METHOD_2.

Alternative input document to validate by customer ID and token (may only be done after initial authentication):

```
<?xml version="1.0"?>
<authentication-request>
  <cust-id>A999999999</cust-id>
  <session-id>Example Hash Code Value</session-id>
</authentication-request>
```

Output Document

```
<?xml version="1.0"?>
<authentication>
  <authenticated>>true</authenticated>
  <authentication-message>Might be for a successful or a failed
authentication</authentication-message>
```

```
<authentication-error-id>11</authentication-error-id>
<session>
  <session-id>Example Hash Code Value</session-id>
  <session_id>Example Hash Code Value</session_id>
  <breadcrumb-session-id>Example Breadcrumb Session ID Value</breadcrumb-session-id>
  <login-serno>1234567</login-serno>
  <roles>
    <role>MEMBER</role>
    <role>GUEST</role>
  </roles>
</session>
<customer>
  <cust-id>A999999999</cust-id>
  <cust-type>I</cust-type>
  <name>
    <display-name>John Doe</display-name>
    <last-name>Doe</last-name>
    <first-name>John</first-name>
    <company-name>That Big University</company-name>
  </name>
  <cust-email>test@hotmail.com</cust-email>
</customer>
<memberships>
  <membership>
    <member>true</member>
    <status>ACTIVE</status>
    <subgroup-ID>ABC</subgroup-ID>
    <subgroup-type>NA</subgroup-type>
    <subgroup-name>Human-Readable Value</subgroup-name>
    <class-code>SAMPLE CODE</class-code>
    <subclass-code>SAMPLE CODE</subclass-code>
    <level-of-service>Sample Description of Level of Service</level-of-service>
    <end-of-service-date>2011-12-31</end-of-service-date>
    <paid-through-date>2010-12-31</paid-through-date>
  </membership>
  <membership>
    ...
  </membership>
</memberships>
<subscriptions>
  <subscription>
    <package-code>SAMPLE CORE PACK CODE</package-code>
    <package-name>Sample Core Package Name</package-name>
    <benefit-of-membership>true</benefit-of-membership>
    <associated-subgroup-ID>ABC</associated-subgroup-ID>
    <end-of-service-date>2011-12-31</end-of-service-date>
    <paid-through-date>2010-12-31</paid-through-date>
  </subscription>
</subscriptions>
  ...
</subscriptions>
```

```
</authentication>
```

General Notes on XML Schema:

1. On failed authentication, only authenticated; authentication-message & authentication-error-id (see Appendix B) will be returned
2. On successful authentication, authenticated; authentication-message; session; customer; memberships & subscriptions will exist.
3. Role, membership, subscription may all have 0, 1, or multiple children, but are likely to have 1 or more.
4. Within Role, membership, subscription all children will exist.
5. Name will always have a display-name child; first-name, last-name, and company-name may or may not exist depending on cust-type.

***ssaauthmain.authenticated_redirect* to perform redirect from partner website to AGCI url with pseudo-single sign on:**

Arguments:

p_cust_id

p_hash_cd (session ID)

p_target_url

p_add_cust_id_to_url (default to Y)

target URL could be constructed, including the customer ID, on partner side, but the easier approach is to use the same syntax as the SSA menus, and let the database add the customer ID.


Appendix A: Data Element Descriptions

Data element	Purpose	Type	Length
Session-id	Hashed Session ID	Hash String	30
Cust-id	Unique Customer ID within ACGI AMS	String	10
Role	Privileges for Internet Users of ACGI (i.e. ABC members/subscribers); probably not used by CJO	String	30
Cust-type	Member Type: will be I (Individual) for now; may be expanded to include C (companies) in the future	String	2
Display-name	Member Name, formatted for display	String	60
Last-name	Member Last Name, if Individual	String	30
First-name	Member First Name, if Individual	String	30
Member	Is current active member for this subgroup; always true for now, reserved for future expansion	Boolean	
[Membership] Status	ACTIVE, SUSPENDED, TERMINATED, PENDING; always ACTIVE for now, reserved for future expansion	String	10


Subgroup-ID	Identifier of a membership subgroup; may be ABC, special interest groups, regional chapters etc. Only ABC membership will matter.	String	10
Subgroup-type	Type of Membership Subgroup, reserved for future expansion. Code values are configurable, but frequently include NA (National Association) and CH (chapter)	String	2
Subgroup-name	Name of Membership Subgroup, formatted for display	String	60
Class-code & Subclass-Code	Two dimensional Codes for differentiating membership privileges and pricing. To be configured by ABC; probably not used by CJO.	String	8
Level-of-service	Description of the combination of class and subclass	String	60
[Membership] End-of-service-date	Date when the membership will expire; reserved for future expansion, probably not used by CJO.	Date String YYYY-MM-DD	10
[Membership] Paid-through-date	Date through which the membership has been paid (usually the same as the end of service date); reserved for future expansion, probably not used by CJO.	Date String YYYY-MM-DD	10
Package-code	Unique Code Identifier for a Set of one or more publications that can be subscribed to as a unit. These will be defined by ABC, and will be needed by CJO to differentiate members with and without online access.	String	30
Package-name	Description of a Subscription Package	String	60
Benefit-of-membership	If a subscription is a free benefit of membership or a separately-paid subscription	boolean	
Associated-subgroup-ID	For benefit of membership subscriptions, the Subgroup for which this is a benefit	String	10
[Subscription] End-of-service-date	Date when the subscription will expire; reserved for future expansion, probably not used by CJO.	Date String YYYY-MM-DD	10
[Subscription] Paid-through-date	Date through which the subscription has been paid (usually the same as the end of service date); reserved for future expansion, probably not used by CJO.	Date String YYYY-MM-DD	10

Appendix B: Anticipated Error Conditions and Coding

The following table defines the error/failure conditions anticipated at this point.

 This list accurate up to AA 6.4.5.

Authentication-error-id	Authentication-message
1	Badly-formed XML
10	Authentication-request not found
20	Invalid xml: neither username nor cust-id found
30	Invalid xml: password not found
40	Invalid xml: session-id not found
100	Bad username/password combination
200	Invalid cust-id
201	Invalid Session ID - session no longer valid
	Successfully Authenticated

 This list changed as of AA 6.5. The new list can be seen below

Authentication-error-id	Authentication-message
1	Badly-formed XML
30	Invalid xml: password not found
50	Invalid xml: no complete authentication details
60	Your IP may not access this function
100	Bad authentication details. This could be due to the system not finding a matching username/password. Or if the username/password exists, the system may not be able to find a valid SSA role for the customer.
200	Invalid cust-id
201	Invalid Session ID - session no longer valid
999	Unexpected error: <i>technical error message will be included here</i>
	Successfully Authenticated