6.4.2.6.6 Get Event Registration Info XML Web Service - EVTSSAWEBSVCLIB. GET_EVENTREG_INFO_XML

Web Service

This web service is available as of AA version 6.5.1

ACGI has developed a web service that allows permitted third-party sites to gather event registration details about a given customer/event but does not require an ODBC connection.

Web service Details

The "get event registration information" 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. ACGI also recommends only requesting the necessary data, to avoid unnecessary overhead. The example below shows how to pull all currently supported data.

See APPENDIX A for Error Conditions and Error Codes

evtssawebsvclib.get_eventreg_info_xml **passing arguments as an xml document with name "p_input_xml_doc"**

Prior to AA 6.5.1, the cust-id element required a not-null value.

In 7.3.4, changes were made to the response to include guests (by name) associated with each registrant.

In 7.4, changes were made to the response to include the event registrant's address detail (e.g., <evt-reg-street1>).

In 7.4.2, added the ability to filter secondary items to ones that required tickets or did not, using the element /eventreg-request /secondaries-require-ticket.

In 7.4.5, added the ability to include item registration attributes in the response.

in 8.1.0, added the ability to include registration attributes if required. (WIP)

in 8.1.3, added <eventStatus> to the response

in AA8.7.0, added URLtoFile to the response attributes

in AA8.7.0, added URLtoThumbnail to the response attributes

in 8.7.4, added the following filters to the request: primaryRegStatus, regOrChangedAfterDate, regOrChangedBeforeDate, and eventType

NOTE: The primary purpose of this approach was to facilitate Higher Logic integration efforts.

Input Document Example

```
<?xml version="1.0" encoding="UTF-8"?>
   <eventreg-request>
           <vendor-id></vendor-id>
           <vendor-password></vendor-password>
           <cust-id></cust-id><!-- Optional, but should not be null if
event-id is null -->
```

```xml
            <event-id></event-id><!-- Optional, but should not be null
if cust-id is null -->
            <first-name></first-name><!-- Optional, used as an
alternate lookup parameter -->
            <last-name></last-name><!-- Optional, used as an alternate
lookup parameter -->
            <company-name></company-name><!-- Optional, used as an
alternate lookup parameter -->
            <email></email><!-- Optional, used as an alternate lookup
parameter -->
            <eventType></eventType><!-- Optional: if used, must also
provide any of the following filters: cust ID, event ID,
regOrChangedBefore, or regOrChangedAfter -->
            <primaryRegStatus></primaryRegStatus><!-- Optional; if not
provided, will return all statuses except "CANCELLED" as before. "ANY"
will return all statuses. -->
            <regOrChangedAfterDate></regOrChangedAfterDate><!--
Optional; dates support timestamps but will also work without them.
supported formats are MM/DD/YYYY and MM/DD/YYYY HH24:MI:SS -->
            <regOrChangedBeforeDate></regOrChangedBeforeDate><!--
Optional; dates support timestamps but will also work without them.
supported formats are MM/DD/YYYY and MM/DD/YYYY HH24:MI:SS -->
            <secondaries-require-ticket></secondaries-require-ticket><!
-- Optional boolean input. Further explanation below -->
            <regAttributes include="true" includeAll="true">
                <!--
                The following are examples of querying registration
attributes by type or type+code.
                Do not include the attr children if includeAll is set
to "true"


                <attr type="CELLPHONE" />
                <attr type="FREE_GIFT" code="TSHIRT" />
                 -->
            </regAttributes>
            <regItemAttributes include="true" includeAll="true">
                <!--
                The following are examples of querying registration
item attributes by type or type+code.
                Do not include the attr children if includeAll is set
to "true"


                <attr type="DIET_RESTRICTION" />
                <attr type="FREE_GIFT" code="TSHIRT" />
                -->
            </regItemAttributes>
    </eventreg-request>
```

Output Document

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<registrations>
    <registration><!-- can repeat -->
        <regi-serno></regi-serno>
        <customer-id></customer-id>
        <event-id></event-id>
        <eventStatus></eventStatus>
        <registration-date></registration-date><!-- MM/DD/YYYY -->
        <registration-type></registration-type>
        <registration-name></registration-name>
        <representing></representing>
        <billto-id></billto-id>
        <promo-cd></promo-cd>
        <purchase-order></purchase-order>
        <prim-item-id></prim-item-id>
        <prim-reg-status></prim-reg-status>
        <total-charges></total-charges>
        <total-payment></total-payment>
        <balance></balance>
        <event-nm></event-nm>
        <program-nm></program-nm>
        <primary-item-descr></primary-item-descr>
        <event-start-dt></event-start-dt>
        <event-end-dt></event-end-dt>
        <location-name></location-name>
        <location-street1></location-street1>
        <location-street2></location-street2>
        <location-city></location-city>
        <location-state></location-state>
        <location-zip></location-zip>
        <location-country></location-country>
        <location-country-descr></location-country-descr>
        <first-name></first-name>
        <last-name></last-name>
        <company-name></company-name>
        <email></email>
        <evt-reg-street1></evt-reg-street1>
        <evt-reg-street2></evt-reg-street2>
        <evt-reg-street3></evt-reg-street3>
        <evt-reg-city></evt-reg-city>
        <evt-reg-state></evt-reg-state>
        <evt-reg-zip></evt-reg-zip>
        <evt-reg-country></evt-reg-country>
        <regAttributes>
            <regAttribute><!-- can repeat -->
                <type><![CDATA[]]></type>
                <typeName><![CDATA[]]></typeName>
                <code><![CDATA[]]></code>
```

```xml
                    <codeDescr><![CDATA[]]></codeDescr>
                    <char><![CDATA[]]></char>
                    <number />
                    <date />
                    <URLtoFile></URLtoFile>           <!-- CDATA wrapped -->
                    <URLtoThumbnail></URLtoThumbnail> <!-- CDATA wrapped -->
                </regAttribute>
            </regAttributes>
            <items>
                <item><!-- can repeat -->
                    <id></id>
                    <descr></descr>
                    <registration-type></registration-type>
                    <registration-status></registration-status>
                    <registration-date></registration-date>
                                <quantity></quantity>
                    <attended></attended><!-- Y/N/Null -->
                    <regItemAttributes>
                        <regItemAttribute><!-- can repeat -->
                            <type><![CDATA[]]></type>
                            <typeName><![CDATA[]]></typeName>
                            <code><![CDATA[]]></code>
                            <codeDescr><![CDATA[]]></codeDescr>
                            <char><![CDATA[]]></char>
                            <number />
                            <date />
                            <URLtoFile></URLtoFile>           <!-- CDATA
wrapped -->
                            <URLtoThumbnail></URLtoThumbnail> <!-- CDATA
wrapped -->
                        </regItemAttribute>
                    </regItemAttributes>
                </item>
            </items>
            <guests>
                <guest><!-- can repeat -->
                    <firstNm></firstNm>
                    <middleNm></middleNm>
                    <lastNm></lastNm>
                </guest>
            </guests>
        </registration>
        <status>SUCCESS or FAILURE</status>
        <message>Element will not appear if status = SUCCESS</message>
    </registrations>
```

Output Document (failure example)

```
<?xml version="1.0" encoding="UTF-8"?>
<errors>
  <error code="1">Badly-formed XML</error>
</errors>
```

Secondary Items Require Ticket Filter

The element /eventreg-request/secondaries-require-ticket supports boolean values (true or false). If "true", then only secondaries set up to require tickets will be returned. If "false", then only secondaries set up to **not** require tickets will be returned. If no value (or anything other than true/false) is provided, then setup to require tickets will have no effect the response.

Appendix A: Error Codes

The following table defines the possible error/failure codes.

error code	error message
1	Badly-formed XML
60	Your IP may not access this function
100	Bad integrator username/password combination
999	Unexpected error: *technical error message will be included here*