

API Inquiry: *get_eventgroup*

Description	Returns all fields from the events groups table for all groups (teams or tables), as well as registration, prospect and e-mail information for all prospects assigned to the group.	
Parameters	scheme	GET <i>Required.</i> All connection types. Inquiry type indicator.
	action	get_eventgroup <i>Required.</i> All connection types. Variable representing the specific inquiry.
	URL	nnn.nnn.n.n <i>Optional.</i> Direct and DAD only. The URL of the requester, used for reporting. Results will display only if a URL is included.
	dsource	xxxxxxx <i>Required.</i> Connection pooling servlet only. The key to the client door to the database.
	datafile	event optional Restrict to event_groups rows with this as the event
		groupname optional Restrict to event_groups rows matching this -- but ignore case and spaces on both sides of the comparison. Do not add any wildcards to the value passed in; if wildcards are desired, they will be passed in.
		groupype optional Restrict to event_groups rows with this as the groupype
		transnum optional Restrict to event_groups row with this as the transnum
		parent_event optional Restrict to event_groups rows where event.parent_event matches this value.
		event_fundcode optional Restrict to event_groups rows where event.fundcode matches this value.
		event_unitid optional Restrict to event_groups rows where event.unitid matches this value.
		event_start_from optional Expect a date. Restrict to event_groups rows where event.start_date is on or after the date passed in.
		nametype optional Restrict the registrations returned for groups. This should not affect which groups are returned -- but only restrict the registrations. If nametype is received, only return for each group those registrations for which there exists a reg_name with this nametype value. The join is reg_name.regid = registration.transnum.

API Inquiry: *get_eventgroup*

		positiontype	<p>optional</p> <p>Restrict the registrations returned for groups. This should not affect which groups are returned -- but only restrict the registrations. If positiontype is received, only return for each group those registrations for which there exists a reg_name assigned to a position of this type (event_positions.position_type). For example, only return registrations for team leaders. The join is reg_name.regid = registration.transnum and reg_name.positionid = event_positions.transnum</p>
Sample Data	<pre><?xml version="1.0" encoding="UTF-8"?> <XMLREQUEST> <RECORD> <EVENT>EVENT</EVENT> <GROUPNAME>Team USA</GROUPNAME> </RECORD> </XMLREQUEST></pre>		
Sample Result	<pre><GET_EVENTGROUP TRANS_TIMESTAMP="10/30/2009 08:54:33 am" EVENT="EVENT" GROUPNAME="Team USA" POSITIONTYPE="POS1"> - <EVENT_GROUPS> <EVENT_GROUPS.TCN>1514795750</EVENT_GROUPS.TCN> <EVENT_GROUPS.GROUPTYPE>1234</EVENT_GROUPS.GROUPTYPE> <EVENT_GROUPS.NUMPOSITIONS>15</EVENT_GROUPS.NUMPOSITIONS> <EVENT_GROUPS.POSITIONTYPE>POS1</EVENT_GROUPS.POSITIONTYPE> <EVENT_GROUPS.GROUPNAME>Team USA</EVENT_GROUPS.GROUPNAME> <EVENT_GROUPS.EVENT>EVENT</EVENT_GROUPS.EVENT> <EVENT_GROUPS.TIMESTAMP>11/24/2004</EVENT_GROUPS.TIMESTAMP> <EVENT_GROUPS.TRANSNUM>2028</EVENT_GROUPS.TRANSNUM> <EVENT_GROUPS.UPDATE_DATE>10/30/2009</EVENT_GROUPS.UPDATE_DATE> <EVENT_GROUPS.USERID_ADD>JBK420</EVENT_GROUPS.USERID_ADD> <EVENT_GROUPS.USERID_CHANGE>TESTV452</EVENT_GROUPS.USERID_CHANGE> - <EV_POS_REG_NAME_PROSPECT_EMAIL> - <ROWSET> - <DETAIL> <EVENT_POSITIONS.POSITIONSEQ>1</EVENT_POSITIONS.POSITIONSEQ> <EVENT_POSITIONS.GROUPID>2028</EVENT_POSITIONS.GROUPID> <EVENT_POSITIONS.POSITIONTYPE>POS1</EVENT_POSITIONS.POSITIONTYPE> <EVENT_POSITIONS.SECTION>15</EVENT_POSITIONS.SECTION> <EVENT_POSITIONS.AISLE>1</EVENT_POSITIONS.AISLE> <EVENT_POSITIONS.SEATNUM>15</EVENT_POSITIONS.SEATNUM> <EVENT_POSITIONS.POSITIONCLASS>1234</EVENT_POSITIONS.POSITIONCLASS> <EVENT_POSITIONS.TIMESTAMP>11/24/2004</EVENT_POSITIONS.TIMESTAMP> <EVENT_POSITIONS.TRANSNUM>17730</EVENT_POSITIONS.TRANSNUM> <EVENT_POSITIONS.USERID_ADD>JBK420</EVENT_POSITIONS.USERID_ADD> <EVENT_POSITIONS.TCN>1</EVENT_POSITIONS.TCN></pre>		

API Inquiry: *get_eventgroup*

```

<REG_NAME.NAME_ID>343632</REG_NAME.NAME_ID>
<REG_NAME.GROUPID>2028</REG_NAME.GROUPID>
<REG_NAME.NAME>Kohl</REG_NAME.NAME>
<REG_NAME.POSITIONID>17730</REG_NAME.POSITIONID>
<REG_NAME.REGID>115817</REG_NAME.REGID>
<REG_NAME.TIMESTAMP>10/20/2006</REG_NAME.TIMESTAMP>
<REG_NAME.TRANSNUM>18364</REG_NAME.TRANSNUM>
<REG_NAME.UPDATE_DATE>10/30/2009</REG_NAME.UPDATE_DATE>
<REG_NAME.USERID_ADD>JBK441</REG_NAME.USERID_ADD>
<REG_NAME.USERID_CHANGE>TESTV452</REG_NAME.USERID_CHANGE>
<REG_NAME.FNAME>Joan</REG_NAME.FNAME>
<REG_NAME.USERTEXT1>jkohl@pledgemaker.com</REG_NAME.USERTEXT1>
<REG_NAME.USERTEXT2>716-6914935</REG_NAME.USERTEXT2>
<REG_NAME.TCN>1514792053</REG_NAME.TCN>
<REGISTRATION.REMARK>FEEQTY: 4</REGISTRATION.REMARK>
<REGISTRATION.USERID_ADD>LOCKBOX</REGISTRATION.USERID_ADD>
<REGISTRATION.IDNUMBER>343632</REGISTRATION.IDNUMBER>
<REGISTRATION.TRANSNUM>115817</REGISTRATION.TRANSNUM>
<REGISTRATION.APPEALCODE>12345678901234567890</REGISTRATION.APPEALCODE>
<REGISTRATION.COMPLIMENTARY>N</REGISTRATION.COMPLIMENTARY>
<REGISTRATION.EVENT>EVENT</REGISTRATION.EVENT>
<REGISTRATION.USERCODE6>N</REGISTRATION.USERCODE6>
<REGISTRATION.USERFLAG1>N</REGISTRATION.USERFLAG1>
<REGISTRATION.UNITID>24929</REGISTRATION.UNITID>
<REGISTRATION.PREMIUM_DECLINED>N</REGISTRATION.PREMIUM_DECLINED>
<REGISTRATION.TCN>1</REGISTRATION.TCN>
<REGISTRATION.TOTAL>0</REGISTRATION.TOTAL>
<REGISTRATION.BALANCE>0</REGISTRATION.BALANCE>
<REGISTRATION.T_DATE>06/02/2006</REGISTRATION.T_DATE>
<REGISTRATION.TIMESTAMP>10/20/2006</REGISTRATION.TIMESTAMP>
<PROSPECT.BILLFLAG>Y</PROSPECT.BILLFLAG>
<PROSPECT.TCN>453251027</PROSPECT.TCN>
<PROSPECT.EMAILFLAG>Y</PROSPECT.EMAILFLAG>
<PROSPECT.IDNUMBER>343632</PROSPECT.IDNUMBER>
<PROSPECT.FIRST>Joan</PROSPECT.FIRST>
<PROSPECT.LAST>Orange</PROSPECT.LAST>
<PROSPECT.SOLICIT>Y</PROSPECT.SOLICIT>
<PROSPECT.MAILFLAG>Y</PROSPECT.MAILFLAG>
<PROSPECT.USERCODE1>C</PROSPECT.USERCODE1>
<PROSPECT.USERCODE2>UN</PROSPECT.USERCODE2>
<PROSPECT.USERCODE3>BRON</PROSPECT.USERCODE3>
<PROSPECT.USERCODE8>MAIL</PROSPECT.USERCODE8>
<PROSPECT.SOURCECODE>APPEAL4</PROSPECT.SOURCECODE>
<PROSPECT.TIMESTAMP>10/20/2006</PROSPECT.TIMESTAMP>
<PROSPECT.USERID_ADD>LOCKBOX</PROSPECT.USERID_ADD>
<PROSPECT.FAMILYLINKTYPE>U</PROSPECT.FAMILYLINKTYPE>
<PROSPECT.RECORDTYPE>I</PROSPECT.RECORDTYPE>
<PROSPECT.GREETINGPREF>S</PROSPECT.GREETINGPREF>
<PROSPECT.GREETING>Orange</PROSPECT.GREETING>
<PROSPECT.CORPORATELINKTYPE>U</PROSPECT.CORPORATELINKTYPE>
<PROSPECT.CT_ACCESS>Y</PROSPECT.CT_ACCESS>
<PROSPECT.ANONYMOUS>N</PROSPECT.ANONYMOUS>

```

API Inquiry: `get_eventgroup`

	<pre> <PROSPECT.BATCHNO>4071</PROSPECT.BATCHNO> <PROSPECT.ISUNIT>N</PROSPECT.ISUNIT> <PROSPECT.UNITID>24929</PROSPECT.UNITID> <PROSPECT.FREQUENCYTYPE>O</PROSPECT.FREQUENCYTYPE> <PROSPECT.JANUARY>N</PROSPECT.JANUARY> <PROSPECT.FEBRUARY>N</PROSPECT.FEBRUARY> <PROSPECT.MARCH>N</PROSPECT.MARCH> <PROSPECT.APRIL>N</PROSPECT.APRIL> <PROSPECT.MAY>N</PROSPECT.MAY> <PROSPECT.JUNE>N</PROSPECT.JUNE> <PROSPECT.JULY>N</PROSPECT.JULY> <PROSPECT.AUGUST>N</PROSPECT.AUGUST> <PROSPECT.SEPTEMBER>N</PROSPECT.SEPTEMBER> <PROSPECT.OCTOBER>N</PROSPECT.OCTOBER> <PROSPECT.NOVEMBER>N</PROSPECT.NOVEMBER> <PROSPECT.DECEMBER>N</PROSPECT.DECEMBER> <EMAIL.TRANSNUM>277</EMAIL.TRANSNUM> <EMAIL.ATYPE>H</EMAIL.ATYPE> <EMAIL.EMAIL>jkohl@pledgemaker.com</EMAIL.EMAIL> <EMAIL.DELIVERABLE>Y</EMAIL.DELIVERABLE> <EMAIL.IDNUMBER>343632</EMAIL.IDNUMBER> <EMAIL.STARTDATE>10/20/2006</EMAIL.STARTDATE> <EMAIL.FORMAT_PREF>B</EMAIL.FORMAT_PREF> <EMAIL.ALLOW_ATTACH>N</EMAIL.ALLOW_ATTACH> <EMAIL.SEASONAL>N</EMAIL.SEASONAL> <EMAIL.PREFERENCE>Y</EMAIL.PREFERENCE> <EMAIL.TIMESTAMP>10/20/2006</EMAIL.TIMESTAMP> <EMAIL.USERID_ADD>LOCKBOX</EMAIL.USERID_ADD> <EMAIL.TCN>1</EMAIL.TCN> - <ALIAS> - <ROWSET> - <DETAIL> <PROSPECT_ALIAS.ALIAS>fake.email@abc.org</PROSPECT_ALIAS.ALIAS> </DETAIL> </ROWSET> </ALIAS> </DETAIL> </ROWSET> </EV_POS_REG_NAME_PROSPECT_EMAIL> </EVENT_GROUPS> </GET_EVENTGROUP> </pre>
Additional Reading	
Reference	

API Inquiry: `get_eventgroup`

Connection Pooling	http://webapps.pledgemaker.com/test/proc?d_source=12345678&scheme=GET&action=get_eventgroup&datafile=
Data Access Descriptor (DAD)	<a 1.0"="" ?><xmlrequest><record><event>event<="" encoding="UTF-8" event><groupname>team="" groupname><="" href="http://pm-devel/pls/testv440/apiwrap.h_traffic?scheme=GET&url=128.168.0.1&action=get_eventgroup&data=<?xml version=" record><="" usa<="" xmlrequest>"="">http://pm-devel/pls/testv440/apiwrap.h_traffic?scheme=GET&url=128.168.0.1&action=get_eventgroup&data=<?xml version="1.0" encoding="UTF-8"?><XMLREQUEST><RECORD><EVENT>EVENT</EVENT><GROUPNAME>TEAM USA</GROUPNAME></RECORD></XMLREQUEST>
Direct Connection	<pre> set serveroutput on; declare vScheme varchar2(100) := 'GET'; vAction varchar2(200) := 'xml_pm.get_eventgroup'; vURL varchar2(100) := '128.168.0.1'; vXML varchar2(10000); rval_clob CLOB; begin owa.num_cgi_vars := 0; vXML := '<?xml version="1.0" encoding="UTF-8"?>' CHR(10) ' <XMLREQUEST>' CHR(10) ' <RECORD>' CHR(10) ' <LAST>Smith</LAST>' CHR(10) ' </RECORD>' CHR(10) ' </XMLREQUEST>'; rval_clob := NULL; apiwrap.all_traffic(vScheme, vAction, vURL, vXML, rval_clob); IF (rval_clob IS NOT NULL) THEN dbms_output.put_line(substr(rval_clob,1,200)); dbms_output.put_line(substr(rval_clob,201,200)); dbms_output.put_line(substr(rval_clob,401,200)); dbms_output.put_line(substr(rval_clob,601,200)); dbms_output.put_line(substr(rval_clob,801,200)); END IF; end; </pre>