

API Inquiry: `get_eventgroupa`

Description	Like <code>get_eventgroup</code> , but returns a limited number of fields.		
Parameters	scheme	GET	<i>Required.</i> All connection types. Inquiry type indicator.
	action	<code>get_eventgroupa</code>	<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.
		grouptype	optional Restrict to event_groups rows with this as the grouptype
		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 <code>reg_name.regid = registration.transnum</code> .
		positiontype	optional 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 <code>reg_name.regid = registration.transnum</code> and <code>reg_name.positionid = event_positions.transnum</code>

API Inquiry: *get_eventgroupa*

Sample Data	<pre><?xml version="1.0" encoding="UTF-8"?> <XMLREQUEST> <RECORD> <EVENT>2008EVENT</EVENT> <GROUPNAME>Team USA</GROUPNAME> </RECORD> </XMLREQUEST></pre>
Sample Result	<pre><?xml version="1.0"?> - <GET_EVENTGROUPA GROUPNAME="Team USA" EVENT="2008EVENT" TRANS_TIMESTAMP="07/21/2014 01:36:07 pm"> - <EVENT_GROUPS> <EVENT_GROUPS.EVENT>2008EVENT</EVENT_GROUPS.EVENT> <EVENT_GROUPS.TRANSNUM>204</EVENT_GROUPS.TRANSNUM> <EVENT_GROUPS.GROUPNAME>Team USA</EVENT_GROUPS.GROUPNAME> <EVENT_GROUPS.GROUPTYPE>RIDE</EVENT_GROUPS.GROUPTYPE> - <EV_POS_REG_NAME_PROSPECT_EMAIL> - <ROWSET> - <DETAIL> <REG_NAME.NAME_ID>123</REG_NAME.NAME_ID> <REG_NAME.GROUPID>204</REG_NAME.GROUPID> <REG_NAME.NAME>Glast</REG_NAME.NAME> <REG_NAME.FNAME>Gfirst </REG_NAME.FNAME> <EMAIL.EMAIL>four@five.com</EMAIL.EMAIL> </DETAIL> - <DETAIL> <REG_NAME.NAME_ID>28000</REG_NAME.NAME_ID> <REG_NAME.GROUPID>204</REG_NAME.GROUPID> <REG_NAME.NAME>Demdinski</REG_NAME.NAME> <REG_NAME.FNAME>Karen </REG_NAME.FNAME> </DETAIL></pre>

API Inquiry: *get_eventgroupa*

	<pre> <DETAIL> <REG_NAME.NAME_ID>296747</REG_NAME.NAME_ID> <REG_NAME.GROUPID>204</REG_NAME.GROUPID> <REG_NAME.NAME>Me</REG_NAME.NAME> <REG_NAME.FNAME>Merge </REG_NAME.FNAME> <EMAIL.EMAIL>jjjj@jjj.jjj</EMAIL.EMAIL> </DETAIL> - <DETAIL> <REG_NAME.NAME_ID>11013397</REG_NAME.NAME_ID> <REG_NAME.GROUPID>204</REG_NAME.GROUPID> <REG_NAME.NAME>Kensington</REG_NAME.NAME> <REG_NAME.FNAME>Costa </REG_NAME.FNAME> </DETAIL> </ROWSET> </EV_POS_REG_NAME_PROSPECT_EMAIL> </EVENT_GROUPS> </GET_EVENTGROUPA> </pre>
Additional Reading	
Reference	

API Inquiry: `get_eventgroupa`

Connection Pooling	http://webapps.pledgemaker.com/test/proc?d_source=12345678&scheme=GET&action=get_eventgroupa&datafile=
Data Access Descriptor (DAD)	<a 1.0"="" ?><xmlrequest><record><event>2008event<="" 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_eventgroupa&data=<?xml version=" record><="" usa<="" xmlrequest>"="">http://pm-devel/pls/testv440/apiwrap.h_traffic?scheme=GET&url=128.168.0.1&action=get_eventgroupa&data=<?xml version="1.0" encoding="UTF-8"?><XMLREQUEST><RECORD><EVENT>2008EVENT</EVENT><GROUPNAME>TEAM USA</GROUPNAME></RECORD></XMLREQUEST>
Direct Connection	<pre> set serveroutput on; declare vScheme varchar2(100) := 'GET'; vAction varchar2(200) := 'xml_pm.get_eventgroupa'; 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) ' <EVENT>2008EVENT</EVENT> <GROUPNAME>TEAM USA</GROUPNAME>' 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>