

API Inquiry: `get_eventfees`

Description	Returns all columns from table event_fees table that match the data file. Returns Total fee count, fee quantity, fee amount and fee balance		
Parameters	scheme	GET	<i>Required.</i> All connection types. Inquiry type indicator.
	action	get_eventfees	<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.
Search Criteria	datafile	Any valid event_fees column	<i>Required at least one.</i>
Sample Data	<pre><?xml version="1.0" encoding="UTF-8"?> <XMLREQUEST> <RECORD> <EVENT>2008EVENT</EVENT> </RECORD> </XMLREQUEST></pre>		
Sample Result	<pre><?xml version="1.0"?> <GET_EVENTFEES EVENT="2008EVENT" TRANS_TIMESTAMP="07/18/2014 02:04:46 pm"> <EVENT_FEES> <EVENT_FEES.FEECODE>DIN</EVENT_FEES.FEECODE> <EVENT_FEES.FEE_DESCRIPTION>Kickoff Dinner</EVENT_FEES.FEE_DESCRIPTION> <EVENT_FEES.DEFAULT_QTY>1</EVENT_FEES.DEFAULT_QTY> <EVENT_FEES.DEFAULT_AMOUNT>100</EVENT_FEES.DEFAULT_AMOUNT> <EVENT_FEES.TIMESTAMP>09/20/2007</EVENT_FEES.TIMESTAMP> <EVENT_FEES.UPDATE_DATE>09/20/2007</EVENT_FEES.UPDATE_DATE> <EVENT_FEES.USERID_ADD>DEVLCURR</EVENT_FEES.USERID_ADD> <EVENT_FEES.USERID_CHANGE>DEVLCURR</EVENT_FEES.USERID_CHANGE> <EVENT_FEES.BENEFIT>80</EVENT_FEES.BENEFIT> <EVENT_FEES.EVENT>2008EVENT</EVENT_FEES.EVENT> <EVENT_FEES.FEEGROUP>FEE1</EVENT_FEES.FEEGROUP> <EVENT_FEES.TRANSNUM>35</EVENT_FEES.TRANSNUM> <EVENT_FEES.REGISTRANT_TYPE>A</EVENT_FEES.REGISTRANT_TYPE> <EVENT_FEES.AUDIENCE_TYPE>A</EVENT_FEES.AUDIENCE_TYPE> <EVENT_FEES.U_STANDARD>Y</EVENT_FEES.U_STANDARD> <EVENT_FEES.U_DISCOUNT>N</EVENT_FEES.U_DISCOUNT> <EVENT_FEES.TCN>1</EVENT_FEES.TCN> <EVENT_FEES.DESCRPTION>Fee Group 1</EVENT_FEES.DESCRPTION> <EVENT_FEES.TYPE>R</EVENT_FEES.TYPE> <EVENT_FEES.TOTAL_FEECOUNT>5</EVENT_FEES.TOTAL_FEECOUNT> <EVENT_FEES.TOTAL_FEEQUANTITY>7</EVENT_FEES.TOTAL_FEEQUANTITY> <EVENT_FEES.TOTAL_FEEAMOUNT>500</EVENT_FEES.TOTAL_FEEAMOUNT> <EVENT_FEES.TOTAL_FEEBALANCE>400</EVENT_FEES.TOTAL_FEEBALANCE> </EVENT_FEES> </GET_EVENTFEES></pre>		

API Inquiry: `get_eventfees`

Additional Reading	
Reference	

API Inquiry: `get_eventfees`

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