



PLATINUM **SOFTWARE**

Grape SQL Function Reference

Table of Contents

1. Grape SQL Functions	Page 3
1.1. API result functions	Page 3
1.2. Data importing functions	Page 3
1.3. JSON helpers	Page 3
1.4. List query	Page 4
1.5. Table Operations	Page 5
1.6. Reports	Page 5
1.7. User related functions	Page 5
1.8. Session related functions	Page 6
1.9. Other utility functions	Page 6

1. GRAPE SQL FUNCTIONS

1.1 API result functions

This functions deal with the creation of standardized API results (in JSON format) to be sent back to the API call. They can be found in [api_result_json.sql](#)

NAME	PARAMETERS	DESCRIPTION
api_result_error	message TEXT code INTEGER info JSON	Returns a standardized JSON error object with stats as "ERROR" and the other fields populated. Example: <pre>{"status": "ERROR", "message": "Message", "code": -2, "error": {}}</pre>
api_error	message TEXT code INTEGER info JSON	Overload for api_result_error
api_error		With no arguments, an "Unknown error" message will be generated
api_error_invalid_input		Similar to calling <code>api_result_error("Invalid input", -2)</code>
api_error_data_not_found		Similar to calling <code>api_result_error("Data not found", -5)</code>
api_success	keys TEXT[] values TEXT[] types TEXT[]	This function will construct a JSON object containing at least one field, "status" with the value "OK". The 3 input parameters should be arrays containing additional keys, values and the associated types (n/i/number/integer, j/json or nothing for text).
api_success	keys TEXT[] values TEXT[]	
api_success	key TEXT value INTEGER	
api_success	key1 TEXT value1 INTEGER key2 TEXT value2 INTEGER	Create an API result success JSON object with two integer fields added.
api_success	key TEXT value JSON	Create an API result success JSON object with a JSON field merged into the result.
api_success		Returns a API result object with a status field set to "OK".

1.2 Data importing functions

NAME	PARAMETERS	DESCRIPTION
data_import_insert		
data_upload_done		
data_import_row_insert		

1.3 JSON helpers

NAME	PARAMETERS	DESCRIPTION
json2xml	data JSON root TEXT	
json_diff	old JSONB new JSONB	
json_diff	old JSON new JSON	

NAME	PARAMETERS	DESCRIPTION
json_object_diff	old JSONB new JSONB	
json_array_diff	old JSONB new JSONB	
json_to_composite_type_text	target_schema TEXT target_type TEXT data JSON	
json_to_composite_type	target_schema TEXT target_type TEXT data JSON	This function will populate a custom type from a JSON object. Multi-level nested objects are supported.
cast_json_array_to_int_array	JSON data	Provides an implicit cast from JSON to INT[] (cast_json_array_to_int_array.sql). <pre># select cast_json_array_to_int_array(' [1,2,3]'::JSON); cast_json_array_to_int_array ----- {1, 2, 3}</pre>
cast_json_array_to_text_array	JSON data	Provides an implicit cast from JSON to TEXT[] (cast_json_array_to_text_array.sql).

1.4 List query

Grape's `list_query` call provides an easy way to retrieve rows from a table. Before the contents of a table can be retrieved this way it needs to be added to a whitelist. This functions can be found in [list_query.sql](#). The built-in API call to access this function is **/grape/list**. Access control is enforced on tables retrieved.

The **grape.list_query** function returns rows from a database table. The following input fields are recognized:

- tablename
- schema (optional) TEXT
- sortfield (optional) TEXT
- sortorder (optional) TEXT DESC
- limit (optional) INTEGER default 50
- offset (optional) INTEGER default 0
- filter (optional) array of fields:
- field TEXT
- operand TEXT of '=', '#x003E;', '#x003C;', '#x003E;=', '#x003C;=', 'LIKE', 'ILIKE', 'IS_NULL', 'IS_NOT_NULL', 'IN'
- value text

The following functions deals with the access control:

NAME	PARAMETERS	DESCRIPTION
grape.list_query_whitelist_add	schema TEXT tables TEXT[] - A list of table names to allow roles TEXT[] - A list of roles to allow	Adds tables to the whitelist for use in grape list_query. Users must be in _roles to be able to access the data in the table.
grape.list_query_whitelist_delete	schema TEXT	Removes a table from the whitelist.

NAME	PARAMETERS	DESCRIPTION
	<i>tablename</i> TEXT - A table to remove from allow	

1.5 Table Operations

Grape provides 3 API calls to perform generic DML (INSERT, UPDATE and DELETE) on whitelisted tables.

The API calls are:

- [GrapeInsertRecord](#)
- [GrapeUpdateRecord](#)
- [GrapeDeleteRecord](#)

The SQL function used to whitelist tables, are *grape.table_operation_whitelist_add(schema TEXT, tables TEXT[], roles TEXT[], allowed_operation TEXT)*.

- *schema* - The schema of the table
- *tables* - An array of table names to add
- *roles* - An array of role names to allow
- *allowed_operation* - The operation to allow (INSERT, UPDATE or DELETE)

1.6 Reports

This functions can be found in [reports.sql](#).

NAME	PARAMETERS	DESCRIPTION
save_report	<i>settings</i> JSON	
execute_report	<i>report_id</i> INTEGER <i>parameters</i> JSON	
execute_report	<i>parameters</i> JSON	

1.7 User related functions

NAME	PARAMETERS	DESCRIPTION
grape.user_save	JSON containing: <i>user_id</i> INTEGER <i>username</i> TEXT <i>password</i> TEXT <i>email</i> TEXT <i>fullnames</i> TEXT <i>active</i> BOOLEAN optional <i>role_names</i> TEXT[] <i>employee_guid</i> GUID	Save a user field, or create a new user. API call: POST /grape/user/save
grape.username	<i>user_id</i> INTEGER	Returns the username for a user ID, or NULL if it does not exist.
grape.user_id_from_name	<i>username</i> TEXT	Returns the user ID for a username, or NULL if it does not exist.
grape.user_id_from_fullnames	<i>fullnames</i> TEXT	Returns the user ID for a user found by fullnames, or NULL if it does not exist.
grape.username_from_fullnames	<i>fullnames</i> TEXT	Returns the username for a user found by fullnames, or NULL if it does not exist.
grape.hash_user_password	<i>user_id</i> INTEGER	Hashes a password for user and updates the user table afterwards. <ul style="list-style-type: none"> • If the hash length is the same as the password length and the password starts with a '\$' sign, it is assumed that the password is already hashed and the update is ignored (return -1)

NAME	PARAMETERS	DESCRIPTION
		<ul style="list-style-type: none"> If grape.setting passwords_hashed isn't true, nothing is done (return -2) On success 0 is returned
grape.hash_user_password	username TEXT	Overload for <i>grape.hash_user_password</i> (user_id INTEGER)

1.8 Session related functions

NAME	PARAMETERS	DESCRIPTION
grape.current_user_roles		Returns a list of all roles the current user belongs to.
grape.current_user_id		Returns the integer value of the current session's "grape.user_id" setting. This is typically set with grape before any API call is called
grape.check_session_access	session_id TEXT - Session ID to check for check_path TEXT - Access path to check check_method TEXT - HTTP method to check (GET/POST)	This function performs access control on an API call (based on the path and session ID). It is automatically called by the express app before any API call is performed: <ol style="list-style-type: none"> Check that the path has access control on it. If it cannot be found, the grape setting default_access_allowed is checked, and if true, access will be granted. If not, it will be denied and code 9 will be returned If the path has a role 'guest' granted access to it, everyone will be allowed (even if the session is invalid) If the session is invalid, access will be denied and code 1 returned If the path has a role 'all', only, and all, valid sessions will be granted access If the user has access granted to the access path's role, access is granted If all the above fails, access is denied with code 2
grape.session_insert		
grape.logout	JSON JSON containing session_id	API call /grape/logout
grape.session_ping	JSON JSON containing session_id	Checks validity of a session and returns a JSON object containing the session's username, user_id, fullnames, email, GUID and user_roles. API Call /grape/session_ping
grape.session_insert	username TEXT password TEXT	This function inserts a new session for a valid username and password provided. API call /grape/logout
grape.		
grape.		

1.9 Other utility functions

NAME	PARAMETERS	DESCRIPTION
month_diff	_d1 DATE _d2 DATE	Returns an integer containing the number of months between the two dates provided. If the first parameter is after the second (higher date), the return value will be negative.
set_value	_name TEXT _value TEXT	Sets the value (insert if new, replace if exist) in the key-value pair table grape.setting returning _value.
get_value	_name TEXT _default_value TEXT	Gets the value for setting _name, and if not found it will return _default_value. Defined in setting.sql
generate_uuid		Generates a unique UUID (for example b1086d35-e973-4356-3adc-2eeb6f4963e2). Defined in uuid.sql
array_lowercase	TEXT[]	
clean_telephone_number	_tel TEXT	
random_string	length INTEGER	Generates a random string of length length. Defined in random_string.sql