Server-side function

DbPoll

This function checks the progress of a database query.

Syntax

```
table dbPoll ( handle queryHandle, int timeout [, bool multipleResults = false ] )
```

OOP Syntax Help! I don't understand this!

> **Method**: *queryHandle:poll(...)*

Required Arguments

- **queryHandle:** A query handle previously returned from dbQuery
- **timeout:** How many milliseconds to wait for a result. Use 0 for an instant response (which may return nil). Use -1 to wait until a result is ready. Note: A wait here will freeze the entire server just like executeSQLQuery

> **Important Note:** It is strongly recommended to use this function in callback, as presented in "This example shows how to obtain data without server freeze."

Optional Arguments

- **multipleResults:** Set to true to enable the return values from multiple queries

Returns

- *nil:* Returns nil if the query results are not yet ready. You should try again in a little while. (If you give up waiting for a result, be sure to call dbFree)
- *false:* Returns false if the query string contained an error, the connection has been lost or the query handle is incorrect. This automatically frees the query handle, so you do not have to call dbFree.
  - This also returns two extra values: (See the example on how the retrieve them)
    - *int:* error code
    - *string* error message
- *table:* Returns a table with the result of the query when the query has successfully completed. This automatically frees the query handle, so you do not have to call dbFree. If multipleResults is set to true, it will first return a table pertaining to one query, followed by the results for that query and so on for the next queries.
  - This also returns extra values (only when multipleResults is set to true):
    - *int:* number of affected rows
    - *int:* last insert id

The table is of the format:

```
{
    { colname1=value1, colname2=value2, ... },
    { colname1=value3, colname2=value4, ... },
    ...
}
```

A subsequent table represents the next row.

> **Note:** If a column contained a number it is returned as a number, this includes things which were stored as string but are numbers such as "1" would be returned as 1.