

Class: Mysql2::Client

Inherits: Object	show all
Defined in: lib/mysql2/client.rb	more...

Direct Known Subclasses

[EM::Client](#)

Instance Attribute Summary

[collapse](#)

[#query_options ⇒ Object](#)

[readonly](#)

Returns the value of attribute query_options.

[#read_timeout ⇒ Object](#)

[readonly](#)

Returns the value of attribute read_timeout.

Class Method Summary

[collapse](#)

[.default_query_options ⇒ Object](#)

[.Mysql2::Client.escape\(string\) ⇒ Object](#)

Escape string so that it may be used in a SQL statement.

[.info ⇒ Object](#)

Returns a string that represents the client library version.

Instance Method Summary

[collapse](#)

[#abandon_results! ⇒ Object](#)

When using MULTI_STATEMENTS support, calling this will throw away any unprocessed results as fast as it can in order to put the connection back into a state where queries can be issued again.

[#affected_rows ⇒ Object](#)

returns the number of rows changed, deleted, or inserted by the last statement if it was an UPDATE, DELETE, or INSERT.

[#async_result ⇒ Object](#)

Returns the result for the last async issued query.

`#automatic_close=(false) ⇒ Object`

Set this to `false` to leave the connection open after it is garbage collected.

`#automatic_close? ⇒ Boolean`

`#close ⇒ nil`

Immediately disconnect from the server; normally the garbage collector will disconnect automatically when a connection is no longer needed.

`#encoding ⇒ Object`

Returns the encoding set on the client.

`#escape(string) ⇒ Object`

Escape string so that it may be used in a SQL statement.

`#info ⇒ Object`

`#initialize(opts = {}) ⇒ Client`

constructor

A new instance of Client.

`#last_id ⇒ Object`

Returns the value generated for an `AUTO_INCREMENT` column by the previous `INSERT` or `UPDATE` statement.

`#more_results? ⇒ Boolean`

Returns true or false if there are more results to process.

`#next_result ⇒ Object`

Fetch the next result set from the server.

`#parse_flags_array(flags, initial = 0) ⇒ Object`

`#ping ⇒ Object`

Checks whether the connection to the server is working.

`#prepare(sql) ⇒ Object`

Create a new prepared statement.

`#query(sql, options = {}) ⇒ Object`

`#query_info ⇒ Object`

`#query_info_string ⇒ Object`

`#reconnect=(true) ⇒ Object`

Enable or disable the automatic reconnect behavior of libmysql.

`#select_db(name) ⇒ Object`

Causes the database specified by `name` to become the default (current) database on the connection specified by `mysql`.

`#server_info ⇒ Object`

Returns a string that represents the server version number.

`#socket` ⇒ Object

`#ssl_cipher` ⇒ Object

`#store_result` ⇒ Object

Return the next result object from a query which yielded multiple result sets.

`#thread_id` ⇒ Object

Returns the thread ID of the current connection.

`#warning_count` ⇒ Object

Constructor Details

`#initialize(opts = {})` ⇒ [Client](#)

[permalink](#)

Returns a new instance of Client

[\[View source\]](#) [\[View on GitHub\]](#)

Instance Attribute Details

`#query_options` ⇒ Object (readonly)

[permalink](#)

Returns the value of attribute query_options

[\[View on GitHub\]](#)

`#read_timeout` ⇒ Object (readonly)

[permalink](#)

Returns the value of attribute read_timeout

[\[View on GitHub\]](#)

Class Method Details

`.default_query_options` ⇒ Object

[permalink](#)

[\[View source\]](#) [\[View on GitHub\]](#)

`.Mysql12::Client.escape(string)` ⇒ Object

[permalink](#)

Escape `string` so that it may be used in a SQL statement. Note that this escape method is not connection encoding aware. If you need encoding support use `Mysql2::Client#escape` instead.

[\[View source\]](#) [\[View on GitHub\]](#)

`.info` ⇒ Object

[permalink](#)

Returns a string that represents the client library version.

[\[View source\]](#) [\[View on GitHub\]](#)

Instance Method Details

`#abandon_results!` ⇒ Object

[permalink](#)

When using MULTI_STATEMENTS support, calling this will throw away any unprocessed results as fast as it can in order to put the connection back into a state where queries can be issued again.

[\[View source\]](#) [\[View on GitHub\]](#)

`#affected_rows` ⇒ Object

[permalink](#)

returns the number of rows changed, deleted, or inserted by the last statement if it was an UPDATE, DELETE, or INSERT.

[\[View source\]](#) [\[View on GitHub\]](#)

`#async_result` ⇒ Object

[permalink](#)

Returns the result for the last async issued query.

[\[View source\]](#) [\[View on GitHub\]](#)

`#automatic_close=(false)` ⇒ Object

[permalink](#)

Set this to `false` to leave the connection open after it is garbage collected. To avoid “Aborted connection” errors on the server, explicitly call `close` when the connection is no longer needed.

See Also:

- <http://dev.mysql.com/doc/en/communication-errors.html>

[\[View source\]](#) [\[View on GitHub\]](#)

#automatic_close? ⇒ Boolean

[permalink](#)

Returns:

- (Boolean)
- (Boolean)

[\[View source\]](#) [\[View on GitHub\]](#)

#close ⇒ nil

[permalink](#)

Immediately disconnect from the server; normally the garbage collector will disconnect automatically when a connection is no longer needed. Explicitly closing this will free up server resources sooner than waiting for the garbage collector.

Returns:

- (nil)

[\[View source\]](#) [\[View on GitHub\]](#)

#encoding ⇒ Object

[permalink](#)

Returns the encoding set on the client.

[\[View source\]](#) [\[View on GitHub\]](#)

#escape(string) ⇒ Object

[permalink](#)

Escape **string** so that it may be used in a SQL statement.

[\[View source\]](#) [\[View on GitHub\]](#)

#info ⇒ Object

[permalink](#)

[\[View source\]](#) [\[View on GitHub\]](#)

#last_id ⇒ Object

[permalink](#)

Returns the value generated for an AUTO_INCREMENT column by the previous INSERT or UPDATE statement.

[\[View source\]](#) [\[View on GitHub\]](#)

#more_results? ⇒ Boolean

[permalink](#)

Returns true or false if there are more results to process.

Returns:

- (Boolean)

[\[View source\]](#) [\[View on GitHub\]](#)

#next_result ⇒ Object

[permalink](#)

Fetch the next result set from the server. Returns nothing.

[\[View source\]](#) [\[View on GitHub\]](#)

#parse_flags_array(flags, initial = 0) ⇒ Object

[permalink](#)

[\[View source\]](#) [\[View on GitHub\]](#)

#ping ⇒ Object

[permalink](#)

Checks whether the connection to the server is working. If the connection has gone down and auto-reconnect is enabled an attempt to reconnect is made. If the connection is down and auto-reconnect is disabled, ping returns an error.

[\[View source\]](#) [\[View on GitHub\]](#)

#prepare(sql) ⇒ Object

[permalink](#)

Create a new prepared statement.

[\[View source\]](#) [\[View on GitHub\]](#)

#query(sql, options = {}) ⇒ Object

[permalink](#)

[\[View source\]](#) [\[View on GitHub\]](#)

#query_info ⇒ Object

[permalink](#)

[\[View source\]](#) [\[View on GitHub\]](#)

#query_info_string ⇒ Object

[permalink](#)

[\[View source\]](#) [\[View on GitHub\]](#)

#reconnect=(true) ⇒ Object

[permalink](#)

Enable or disable the automatic reconnect behavior of libmysql. Read dev.mysql.com/doc/refman/5.5/en/auto-reconnect.html for more information.

[\[View source\]](#) [\[View on GitHub\]](#)

#select_db(name) ⇒ Object

[permalink](#)

Causes the database specified by **name** to become the default (current) database on the connection specified by mysql.

[\[View source\]](#) [\[View on GitHub\]](#)

#server_info ⇒ Object

[permalink](#)

Returns a string that represents the server version number

[\[View source\]](#) [\[View on GitHub\]](#)

#socket ⇒ Object

[permalink](#)

[\[View source\]](#) [\[View on GitHub\]](#)

#ssl_cipher ⇒ Object

[permalink](#)

[\[View source\]](#) [\[View on GitHub\]](#)

#store_result ⇒ Object

[permalink](#)

Return the next result object from a query which yielded multiple result sets.

[\[View source\]](#) [\[View on GitHub\]](#)

#thread_id ⇒ Object

[permalink](#)

Returns the thread ID of the current connection.

[\[View source\]](#) [\[View on GitHub\]](#)

#warning_count ⇒ Object

[permalink](#)

[\[View source\]](#) [\[View on GitHub\]](#)

Generated on Tue Aug 16 16:20:05 2016 by yard 0.9.5 (ruby-2.3.1).