

Name

mysql_real_connect - establishes a connection to a MariaDB database server

Synopsis

```
MYSQL * mysql_real_connect(MYSQL *mysql,
                           const char *host,
                           const char *user,
                           const char *passwd,
                           const char *db,
                           unsigned int port,
                           const char *unix_socket,
                           unsigned long flags);
```

Description

Establishes a connection to a database server.

Parameter

- ⊕ mysql - a mysql handle, which was previously allocated by **mysql_init(3)**
- ⊕ host - can be either a host name or an IP address. Passing the NULL value or the string "localhost" to this parameter, the local host is assumed. When possible, pipes will be used instead of the TCP/IP protocol. Since version 3.3.0 it is also possible to provide a comma separated list of hosts for simple fail over in case of one or more hosts are not available.
- ⊕ user - the user name.
- ⊕ passwd - If provided or NULL, the server will attempt to authenticate the user against those user records which have no password only. This allows one username to be used with different permissions (depending on if a password as provided or not).
- ⊕ db - if provided will specify the default database to be used when performing queries.
- ⊕ port - specifies the port number to attempt to connect to the server.
- ⊕ unix_socket - specifies the socket or named pipe that should be used.
- ⊕ flags - the flags allows various connection options to be set

Flag	Description
CLIENT_FOUND_ROWS	Return the number of matched rows instead of number of changed rows.
CLIENT_NO_SCHEMA	Forbids the use of database.tablename.column syntax and forces the SQL parser to generate an error.
CLIENT_COMPRESS	Use compression protocol
CLIENT_IGNORE_SPACE	Allows spaces after function names. All function names will become reserved words.
CLIENT_LOCAL_FILES	Allows LOAD DATA LOCAL statements
CLIENT_MULTI_STATEMENTS	Allows the client to send multiple statements in one command. Statements will be divided by a semicolon.
CLIENT_MULTI_RESULTS	Indicates that the client is able to handle multiple result sets from stored procedures or multi statements. This option will be automatically set if CLIENT_MULTI_STATEMENTS is set.
CLIENT_REMEMBER_OPTIONS	Remembers options passed to mysql_optionsv(3) if a connect attempt failed. If MYSQL_OPTIONS_RECONNECT option was set to true, options will be saved and used for reconnection.

Return value

returns a connection handle (same as passed for 1st parameter) or NULL on error. On error, please check **mysql_errno(3)** and **mysql_error(3)** functions for more information.

Notes

- ⊕ The password doesn't need to be encrypted before executing `mysql_real_connect()`. This will be handled in the client server protocol.
- ⊕ The connection handle can't be reused for establishing a new connection. It must be closed and reinitialized before.

- ⊕ mysql_real_connect() must complete successfully before you can execute any other API functions beside **mysql_optionsv(3)**.
- ⊕ host parameter may contain multiple host/port combinations (supported since version 3.3.0). The following syntax is required:
 - ⊕ hostname and port must be separated by a colon (:)
 - ⊕ IPv6 addresses must be enclosed within square brackets
 - ⊕ hostname:port pairs must be separated by a comma (,)
 - ⊕ if only one host:port was specified, the host string needs to end with a comma.
 - ⊕ if no port was specified, the default port will be used.

Examples for failover host string:

host=[::1]:3306,192.168.0.1:3306,test.example.com

host=127.0.0.1:3306,

See also

- ⊕ **mysql_init(3)**
- ⊕ **mysql_close(3)**
- ⊕ **mariadb_reconnect(3)**
- ⊕ **mysql_error(3)**
- ⊕ **mysql_errno(3)**