## Mysql包

Splits the data into packets of size (2**24**–1) bytes

**packet header**

**每次最大**16MByte

3 1 string[len]

**length**

**sequence\_id**  从0开始，自增

**payload**  [string.fix\_len](http://dev.mysql.com/doc/internals/en/overview.html#type-string.fix_len)

COM\_QUIT

01 00 00 00 01 -- length=0x01, sequence\_id=0x00, command-byte=0x01

## 回包

[OK\_Packet](http://dev.mysql.com/doc/internals/en/overview.html#packet-OK_Packet)

[ERR\_Packet](http://dev.mysql.com/doc/internals/en/overview.html#packet-ERR_Packet)

[EOF\_Packet](http://dev.mysql.com/doc/internals/en/overview.html#packet-EOF_Packet)

### OK\_Packet:

1 [00] the OK header

lenenc-int affected rows

lenenc-int last-insert-id

if capabilities & CLIENT\_PROTOCOL\_41 {

2 status\_flags

2 warnings

} elseif capabilities & CLIENT\_TRANSACTIONS {

2 status\_flags

}

string[EOF] info

# define CLIENT\_PROTOCOL\_41 0x00000200

**Fields**

**header** ([1](http://dev.mysql.com/doc/internals/en/overview.html#type-1)) -- OK header indicator

**affected\_rows** ([lenenc\_int](http://dev.mysql.com/doc/internals/en/overview.html" \l "type-lenenc_int)) -- rows affected by the command

**last\_insert\_id** ([lenenc\_int](http://dev.mysql.com/doc/internals/en/overview.html" \l "type-lenenc_int)) -- last insert-id generated by the command

**status\_flags** ([2](http://dev.mysql.com/doc/internals/en/overview.html#type-2)) -- [status flags](http://dev.mysql.com/doc/internals/en/overview.html#status-flags)

|  |  |
| --- | --- |
| **Status Flag** | **value** |
| **SERVER\_STATUS\_IN\_TRANS** | 0x0001 |
| **SERVER\_STATUS\_AUTOCOMMIT** | 0x0002 |
| **SERVER\_MORE\_RESULTS\_EXISTS** | 0x0008 |
| **SERVER\_STATUS\_NO\_GOOD\_INDEX\_USED** | 0x0010 |
| **SERVER\_STATUS\_NO\_INDEX\_USED** | 0x0020 |
| **SERVER\_STATUS\_CURSOR\_EXISTS** | 0x0040 |
| **SERVER\_STATUS\_LAST\_ROW\_SENT** | 0x0080 |
| **SERVER\_STATUS\_DB\_DROPPED** | 0x0100 |
| **SERVER\_STATUS\_NO\_BACKSLASH\_ESCAPES** | 0x0200 |
| **SERVER\_STATUS\_METADATA\_CHANGED** | 0x0400 |
| **SERVER\_QUERY\_WAS\_SLOW** | 0x0800 |
| **SERVER\_PS\_OUT\_PARAMS** | 0x1000 |

Eg:

07 00 00 02 00 00 00 02 00 00 00 ...........

### ERR\_Packet

1 [ff] the ERR header

2 error code

if capabilities & CLIENT\_PROTOCOL\_41 {

string[1] '#' the sql-state marker

string[5] sql-state

}

string[EOF] error-message

header (1) -- ERR Packet indicator

error\_code (2) -- error-code

sql\_state\_marker (string.fix\_len) -- [len = 1], #

sql\_state (string.fix\_len) -- [len = 5], SQL State of this error

error\_message (string.EOF) -- human readable error message

eg:

17 00 00 01 ff 48 04 23 48 59 30 30 30 4e 6f 20 .....H.#HY000No

74 61 62 6c 65 73 20 75 73 65 64 tables used

### EOF\_Packet

**Caution**the EOF packet may appear in places where a [**Protocol::LengthEncodedInteger**](http://dev.mysql.com/doc/internals/en/overview.html#packet-Protocol::LengthEncodedInteger) may appear. You have to check the packet length is less then 9 to make sure it is a EOF packet.

1 [fe] the EOF header

if capabilities & CLIENT\_PROTOCOL\_41 {

2 warning count

2 status flags

}

header (1) -- EOF Packet indicator

warning\_count (2) -- number of warnings

status\_flags (2) -- status of the statement, see Status Flags

eg:

05 00 00 05 fe 00 00 02 00

## Character Set

SELECT id, collation\_name FROM information\_schema.collations ORDER BY id;

| **number** | **hex** | **character set name** |
| --- | --- | --- |
| 8 | 0x08 | latin1\_swedish\_ci |
| 33 | 0x21 | utf8\_general\_ci |
| 63 | 0x3f | binary |

整型处理：**charset\_nr** ([2](http://dev.mysql.com/doc/internals/en/overview.html#type-2)) -- number of the character set and collation

## Connection Lifecycle

### The MySQL Protocol:

**Connection phase**: a command packet with the sequence-id [00]:

13 00 00 00 03 53 ...

01 00 00 00 01

^^- command-byte

^^---- sequence-id == 0

**Command phase**

包的第一个字节 command-type

| **hex** | **constant name** |
| --- | --- |
| 00 | [**COM\_SLEEP**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_SLEEP) |
| 01 | [**COM\_QUIT**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_QUIT) |
| 02 | [**COM\_INIT\_DB**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_INIT_DB) |
| 03 | [**COM\_QUERY**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_QUERY) |
| 04 | [**COM\_FIELD\_LIST**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_FIELD_LIST) |
| 05 | [**COM\_CREATE\_DB**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_CREATE_DB) |
| 06 | [**COM\_DROP\_DB**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_DROP_DB) |
| 07 | [**COM\_REFRESH**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_REFRESH) |
| 08 | [**COM\_SHUTDOWN**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_SHUTDOWN) |
| 09 | [**COM\_STATISTICS**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_STATISTICS) |
| 0a | [**COM\_PROCESS\_INFO**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_PROCESS_INFO) |
| 0b | [**COM\_CONNECT**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_CONNECT) |
| 0c | [**COM\_PROCESS\_KILL**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_PROCESS_KILL) |
| 0d | [**COM\_DEBUG**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_DEBUG) |
| 0e | [**COM\_PING**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_PING) |
| 0f | [**COM\_TIME**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_TIME) |
| 10 | [**COM\_DELAYED\_INSERT**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_DELAYED_INSERT) |
| 11 | [**COM\_CHANGE\_USER**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_CHANGE_USER) |
| 12 | [**COM\_BINLOG\_DUMP**](http://dev.mysql.com/doc/internals/en/replication-protocol.html#packet-COM_BINLOG_DUMP) |
| 13 | [**COM\_TABLE\_DUMP**](http://dev.mysql.com/doc/internals/en/replication-protocol.html#packet-COM_TABLE_DUMP) |
| 14 | [**COM\_CONNECT\_OUT**](http://dev.mysql.com/doc/internals/en/replication-protocol.html#packet-COM_CONNECT_OUT) |
| 15 | [**COM\_REGISTER\_SLAVE**](http://dev.mysql.com/doc/internals/en/replication-protocol.html#packet-COM_REGISTER_SLAVE) |
| 16 | [**COM\_STMT\_PREPARE**](http://dev.mysql.com/doc/internals/en/prepared-statements.html#packet-COM_STMT_PREPARE) |
| 17 | [**COM\_STMT\_EXECUTE**](http://dev.mysql.com/doc/internals/en/prepared-statements.html#packet-COM_STMT_EXECUTE) |
| 18 | [**COM\_STMT\_SEND\_LONG\_DATA**](http://dev.mysql.com/doc/internals/en/prepared-statements.html#packet-COM_STMT_SEND_LONG_DATA) |
| 19 | [**COM\_STMT\_CLOSE**](http://dev.mysql.com/doc/internals/en/prepared-statements.html#packet-COM_STMT_CLOSE) |
| 1a | [**COM\_STMT\_RESET**](http://dev.mysql.com/doc/internals/en/prepared-statements.html#packet-COM_STMT_RESET) |
| 1b | [**COM\_SET\_OPTION**](http://dev.mysql.com/doc/internals/en/stored-procedures.html#packet-COM_SET_OPTION) |
| 1c | [**COM\_STMT\_FETCH**](http://dev.mysql.com/doc/internals/en/stored-procedures.html#packet-COM_STMT_FETCH) |
| 1d | [**COM\_DAEMON**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_DAEMON) |
| 1e | [**COM\_BINLOG\_DUMP\_GTID**](http://dev.mysql.com/doc/internals/en/replication-protocol.html#packet-COM_BINLOG_DUMP_GTID) |

Command 类型：

**字符协议： COM\_SLEEP COM\_QUIT COM\_INIT\_DB   
 COM\_QUERY**

[Prepared Statements](http://dev.mysql.com/doc/internals/en/prepared-statements.html#section-prepared-statements)  
[Stored Procedures](http://dev.mysql.com/doc/internals/en/stored-procedures.html#section-stored-procedures)

复制

## Connection Phase

[**15.2.1. Initial Handshake**](http://dev.mysql.com/doc/internals/en/connection-phase.html#initial-handshake)

[**15.2.2. Auth Phase Fast Path**](http://dev.mysql.com/doc/internals/en/connection-phase.html#auth-phase-fast-path)

[**15.2.3. Authentication Method Mismatch**](http://dev.mysql.com/doc/internals/en/connection-phase.html#authentication-method-mismatch)

[**15.2.4. Authentication after COM\_CHANGE\_USER command**](http://dev.mysql.com/doc/internals/en/connection-phase.html#authentication-after-com-change-user-command)

[**15.2.5. Connection Phase Packets**](http://dev.mysql.com/doc/internals/en/connection-phase.html#connection-phase-packets)

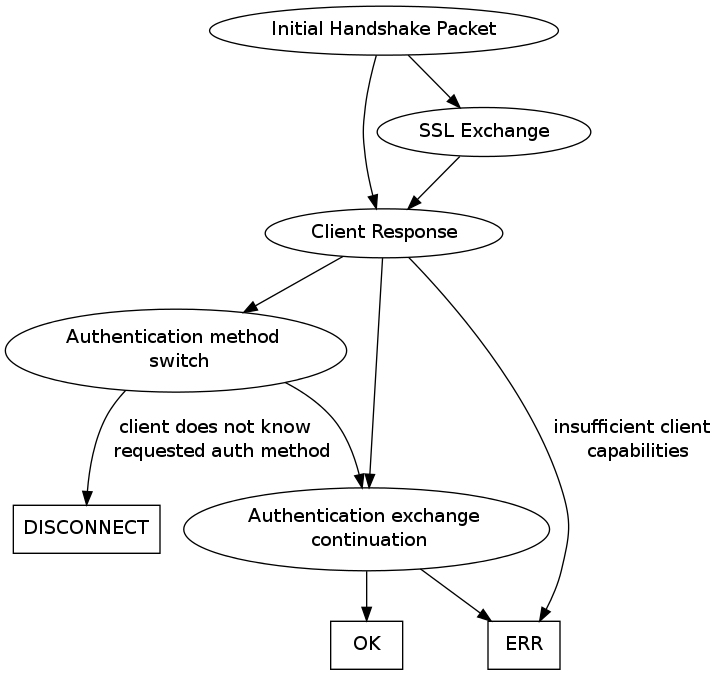
[**15.2.6. Capability Flags**](http://dev.mysql.com/doc/internals/en/connection-phase.html#capability-flags)

The Connection Phase performs these tasks:

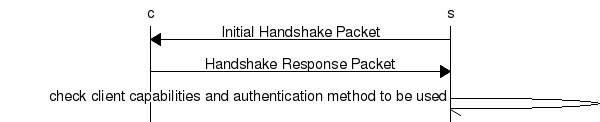
* exchange the capabilities of client and server
* setup SSL communication channel if requested
* authenticate the client against the server

It starts with an initial handshake consisting of server sending [**Initial Handshake Packet**](http://dev.mysql.com/doc/internals/en/connection-phase.html#packet-Protocol::Handshake) and reading clients [**Handshake Response Packet**](http://dev.mysql.com/doc/internals/en/connection-phase.html#packet-Protocol::HandshakeResponse). At this stage client can request SSL connection, in which case an SSL communication channel is established before client sends its authentication response.

After initial handshake, server informs client about the method to be used for authentication (unless it was already established during the handshake) and the authentication exchange continues until server either accepts connection by sending an **[OK\_Packet](http://dev.mysql.com/doc/internals/en/overview.html" \l "packet-OK_Packet)** or rejects it with **[ERR\_Packet](http://dev.mysql.com/doc/internals/en/overview.html" \l "packet-ERR_Packet)**.



|  |  |  |
| --- | --- | --- |
| Request | Response | comment |
| Initial Handshake Packet  1 protocol\_version  ...  1 [09] protocol\_version  string[NUL] server\_version  4 connection\_id  string[NUL] scramble | 4 capability flags, CLIENT\_PROTOCOL\_41 always set  4 max-packet size  1 character set  string[23] reserved (all [0])  string[NUL] username  if capabilities & CLIENT\_PLUGIN\_AUTH\_LENENC\_CLIENT\_DATA {  lenenc-int length of auth-response  string[n] auth-response  } else if capabilities & CLIENT\_SECURE\_CONNECTION {  1 length of auth-response  string[n] auth-response  } else {  string[NUL] auth-response  }  if capabilities & CLIENT\_CONNECT\_WITH\_DB {  string[NUL] database  }  if capabilities & CLIENT\_PLUGIN\_AUTH {  string[NUL] auth plugin name  }  if capabilities & CLIENT\_CONNECT\_ATTRS {  lenenc-int length of all key-values  lenenc-str key  lenenc-str value  if-more data in 'length of all key-values', more keys and value pairs  } | HandshakeV9 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |



各版本兼容性处理

| [CLIENT\_PROTOCOL\_41](http://dev.mysql.com/doc/internals/en/connection-phase.html#flag-CLIENT_PROTOCOL_41) | **no** | **yes** | **yes** | **yes** |
| --- | --- | --- | --- | --- |
| [CLIENT\_SECURE\_CONNECTION](http://dev.mysql.com/doc/internals/en/connection-phase.html#flag-CLIENT_SECURE_CONNECTION) | **--** | **no** | **yes** | **yes** |
| [CLIENT\_PLUGIN\_AUTH](http://dev.mysql.com/doc/internals/en/connection-phase.html#flag-CLIENT_PLUGIN_AUTH) | **--** | **--** | **no** | **yes** |
| [Old Password Authentication](http://dev.mysql.com/doc/internals/en/authentication-method.html#packet-Authentication::Old) | **x** | **x** | **x** | **x** |
| [Secure Password Authentication](http://dev.mysql.com/doc/internals/en/authentication-method.html#packet-Authentication::Native41) |  |  | **x** | **x** |
| Pluggable Authentication |  |  |  | **x** |

#### Successful authentication

A succesful fast authentication path looks as follows:

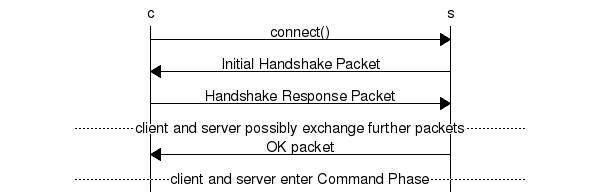
the client connecting to the server

the server responds with the Initial Handshake Packet using auth method M

the client sends the Handshake Response Packet using the same method M

client and server possibly exchange further packets as required by authentication method M

the server responds with OK\_Packet



#### Authentication fails

Server indicates that client is not allowed to connect by sending ERR\_Packet. This can happen at any moment after initial handshake.

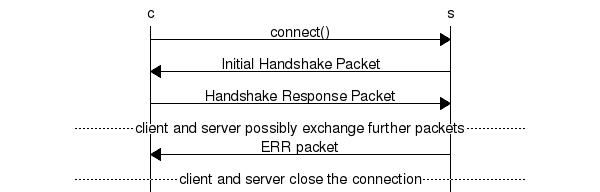
the client connecting to the server

the server responds with the Initial Handshake Packet

the client sends the Handshake Response Packet

client and server possibly exchange further packets as required by the authentication method used

the server responds with ERR\_Packet and closes connection



#### Authentication Method Mismatch

略

## Connection Phase Packets

**HandshakeV10**

1 [0a] protocol version

string[NUL] server version

4 connection id

string[8] auth-plugin-data-part-1

1 [00] filler

2 capability flags (lower 2 bytes)

if more data in the packet:

1 character set

2 status flags

2 capability flags (upper 2 bytes)

if capabilities & CLIENT\_PLUGIN\_AUTH {

1 length of auth-plugin-data

} else {

1 [00]

}

string[10] reserved (all [00])

if capabilities & CLIENT\_SECURE\_CONNECTION {

string[$len] auth-plugin-data-part-2 ($len=MAX(13, length of auth-plugin-data - 8))

if capabilities & CLIENT\_PLUGIN\_AUTH {

if version >= (5.5.7 and < 5.5.10) or (>= 5.6.0 and < 5.6.2) {

string[EOF] auth-plugin name

} elseif version >= 5.5.10 or >= 5.6.2 {

string[NUL] auth-plugin name

}

}

Fields

protocol\_version (1) -- 0x0a protocol\_version

server\_version (string.NUL) -- human-readable server version

connection\_id (4) -- connection id

auth\_plugin\_data\_part\_1 (string.fix\_len) -- [len=8] first 8 bytes of the auth-plugin data

filler\_1 (1) -- 0x00

capability\_flag\_1 (2) -- lower 2 bytes of the Protocol::CapabilityFlags (optional)

character\_set (1) -- default server character-set, only the lower 8-bits Protocol::CharacterSet (optional)

status\_flags (2) -- Protocol::StatusFlags (optional)

capability\_flags\_2 (2) -- upper 2 bytes of the Protocol::CapabilityFlags (since MySQL Server 5.5.7)

auth\_plugin\_data\_len (1) -- length of the combined auth\_plugin\_data, if auth\_plugin\_data\_len is > 0

auth\_plugin\_name (string.NUL) -- name of the auth\_method that the auth\_plugin\_data belongs to

Note

Due to Bug#59453 the auth-plugin-name is missing the terminating NUL-char in versions prior to 5.5.10 and 5.6.2.

**Returns**

[**Protocol::HandshakeResponse**](http://dev.mysql.com/doc/internals/en/connection-phase.html#packet-Protocol::HandshakeResponse) from the client

[**send\_server\_handshake\_packet()**](http://dev.mysql.com/doc/internals/en/source-code-locations.html#send_server_handshake_packet)

example

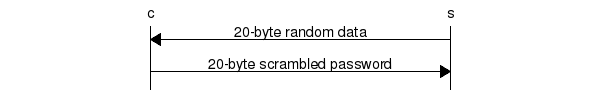
36 00 00 00 0a 35 2e 35 2e 32 2d 6d 32 00 0b 00 6....5.5.2-m2...

00 00 64 76 48 40 49 2d 43 4a 00 ff f7 08 02 00 ..dvH@I-CJ......

00 00 00 00 00 00 00 00 00 00 00 00 00 2a 34 64 .............\*4d

7c 63 5a 77 6b 34 5e 5d 3a 00 |cZwk4^]:.

## Secure Password Authentication



A [**COM\_QUERY**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-COM_QUERY) for **select "012345678901234567890123456789012345"** without [**CLIENT\_COMPRESS**](http://dev.mysql.com/doc/internals/en/connection-phase.html#flag-CLIENT_COMPRESS) has a**payload length** of 46 bytes looks like:

2e 00 00 00 03 73 65 6c 65 63 74 20 22 30 31 32 .....select "012

33 34 35 36 37 38 39 30 31 32 33 34 35 36 37 38 3456789012345678

39 30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 9012345678901234

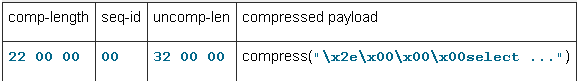
35 22

with [**CLIENT\_COMPRESS**](http://dev.mysql.com/doc/internals/en/connection-phase.html#flag-CLIENT_COMPRESS) the packet is:

22 00 00 00 32 00 00 78 9c d3 63 60 60 60 2e 4e "...2..x..c```.N

cd 49 4d 2e 51 50 32 30 34 32 36 31 35 33 b7 b0 .IM.QP20426153..

c4 cd 52 02 00 0c d1 0a 6c ..R.....l



The compressed packet is 41 bytes long and splits into:

raw packet length -> 41

compressed payload length = 22 00 00 -> 34 (41 - 7)

sequence id = 00 -> 0

uncompressed payload length = 32 00 00 -> 50

Executing **SELECT repeat("a", 50)** results in uncompressed **[ProtocolText::Resultset](http://dev.mysql.com/doc/internals/en/text-protocol.html" \l "packet-ProtocolText::Resultset)** like:

01 00 00 01 01 25 00 00 02 03 64 65 66 00 00 00 .....%....def...

0f 72 65 70 65 61 74 28 22 61 22 2c 20 35 30 29 .repeat("a", 50)

00 0c 08 00 32 00 00 00 fd 01 00 1f 00 00 05 00 ....2...........

00 03 fe 00 00 02 00 33 00 00 04 32 61 61 61 61 .......3...2aaaa

61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 aaaaaaaaaaaaaaaa

61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 aaaaaaaaaaaaaaaa

61 61 61 61 61 61 61 61 61 61 61 61 61 61 05 00 aaaaaaaaaaaaaa..

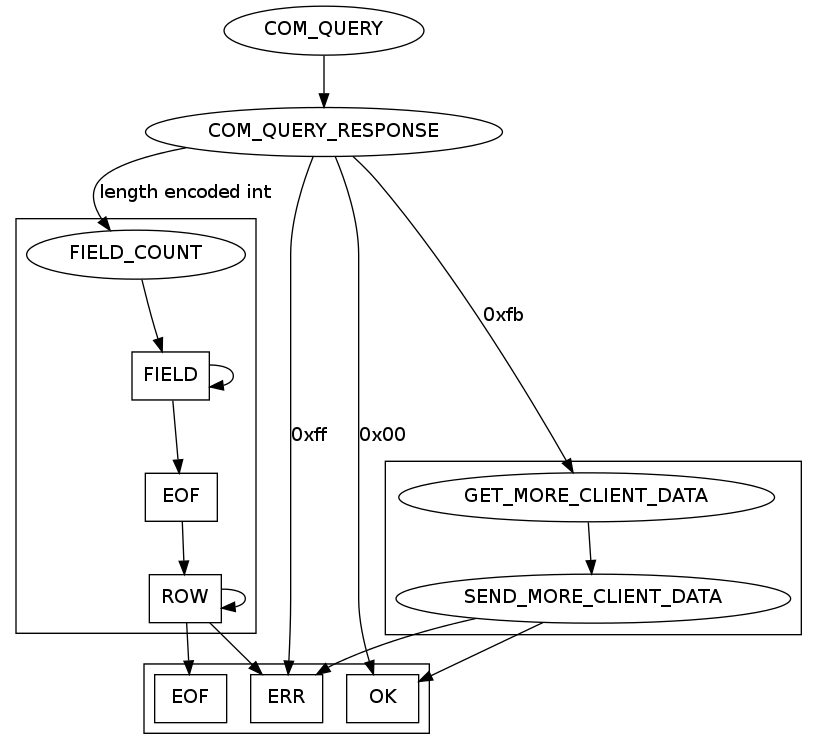
00 05 fe 00 00 02 00 .......

which consists of 5 [**Protocol::Packet**](http://dev.mysql.com/doc/internals/en/overview.html#packet-Protocol::Packet).

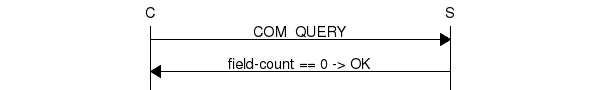
## Text Protocol

|  |  |  |  |
| --- | --- | --- | --- |
|  | Payload | **Returns** | example |
| COM\_SLEEP | 1 [00] COM\_SLEEP | [ERR\_Packet](http://dev.mysql.com/doc/internals/en/overview.html#packet-ERR_Packet) |  |
| COM\_QUIT | 1 [01] COM\_QUIT |  | 01 00 00 00 01 |
| COM\_INIT\_DB | 1 [02] COM\_INIT\_DB | **OK\_Packet** or **ERR\_Packet** | 05 00 00 00 02 74 65 73 74 .....test |
|  | string[EOF] schema name |  |  |
| COM\_QUERY | 1 [03] COM\_QUERY |  |  |
|  | string[EOF] the query the server shall execute |  | [mysql\_query()](http://dev.mysql.com/doc/refman/5.1/en/mysql-query.html) |
| COM\_FIELD\_LIST |  |  |  |
| COM\_CREATE\_DB |  |  |  |
| COM\_DROP\_DB |  |  |  |
| COM\_REFRESH |  |  |  |
| COM\_SHUTDOWN |  |  |  |
| COM\_STATISTICS |  |  |  |
| COM\_PROCESS\_INFO |  |  |  |
| COM\_CONNECT |  |  |  |
| COM\_PROCESS\_KILL |  |  |  |
| COM\_DEBUG |  |  |  |
| COM\_PING |  |  |  |
| COM\_TIME |  |  |  |
| COM\_DELAYED\_INSERT |  |  |  |
| COM\_CHANGE\_USER |  |  |  |
| COM\_DAEMON |  |  |  |

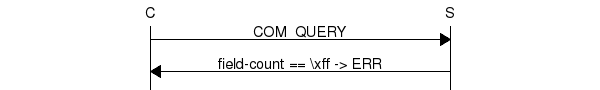
#### COM\_QUERY Response



If the number of columns in the resultset is 0, this is a **[OK\_Packet](http://dev.mysql.com/doc/internals/en/overview.html" \l "packet-OK_Packet)**.



If it is not a valid [**Protocol::LengthEncodedInteger**](http://dev.mysql.com/doc/internals/en/overview.html#packet-Protocol::LengthEncodedInteger) it is either a **[ERR\_Packet](http://dev.mysql.com/doc/internals/en/overview.html" \l "packet-ERR_Packet)** or a[**Protocol::LOCAL\_INFILE\_Request**](http://dev.mysql.com/doc/internals/en/text-protocol.html#packet-Protocol::LOCAL_INFILE_Request).



## Text Resultset

二部分：

列定义

行（值）

**ProtocolText::Resultset:**

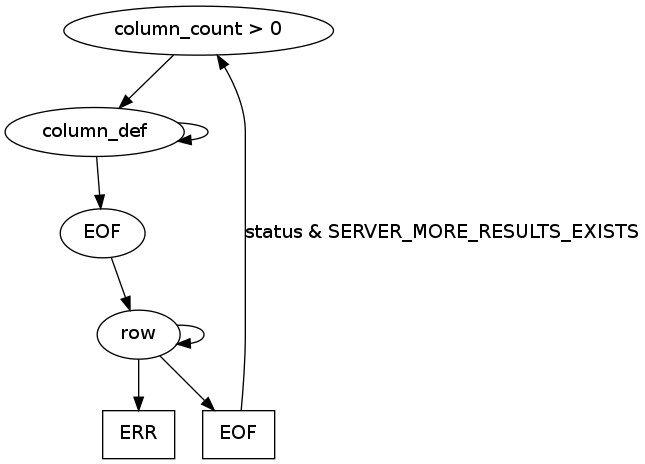
a packet containing a Protocol::LengthEncodedInteger column\_count

column\_count \* Protocol::ColumnDefinition packets

EOF\_Packet

one or more ProtocolText::ResultsetRow packets, each containing column\_count values

EOF\_Packet or ERR\_Packet



**ColumnType**

|  |  |  |
| --- | --- | --- |
|  | **Value** |  |
| **MYSQL\_TYPE\_DECIMAL** | **0x00** |  |
| **MYSQL\_TYPE\_TINY** | **0x01** |  |
| **MYSQL\_TYPE\_SHORT** | **0x02** |  |
| **MYSQL\_TYPE\_LONG** | **0x03** |  |
| **MYSQL\_TYPE\_FLOAT** | **0x04** |  |
| **MYSQL\_TYPE\_DOUBLE** | **0x05** |  |
| **MYSQL\_TYPE\_NULL** | **0x06** |  |
| **MYSQL\_TYPE\_TIMESTAMP** | **0x07** |  |
| **MYSQL\_TYPE\_LONGLONG** | **0x08** |  |
| **MYSQL\_TYPE\_INT24** | **0x09** |  |
| **MYSQL\_TYPE\_DATE** | **0x0a** |  |
| **MYSQL\_TYPE\_TIME** | **0x0b** |  |
| **MYSQL\_TYPE\_DATETIME** | **0x0c** |  |
| **MYSQL\_TYPE\_YEAR** | **0x0d** |  |
| **MYSQL\_TYPE\_NEWDATE** | **0x0e** |  |
| **MYSQL\_TYPE\_VARCHAR** | **0x0f** |  |
| **MYSQL\_TYPE\_BIT** | **0x10** |  |
| **MYSQL\_TYPE\_NEWDECIMAL** | **0xf6** |  |
| **MYSQL\_TYPE\_ENUM** | **0xf7** |  |
| **MYSQL\_TYPE\_SET** | **0xf8** |  |
| **MYSQL\_TYPE\_TINY\_BLOB** | **0xf9** |  |
| **MYSQL\_TYPE\_MEDIUM\_BLOB** | **0xfa** |  |
| **MYSQL\_TYPE\_LONG\_BLOB** | **0xfb** |  |
| **MYSQL\_TYPE\_BLOB** | **0xfc** |  |
| **MYSQL\_TYPE\_VAR\_STRING** | **0xfd** |  |
| **MYSQL\_TYPE\_STRING** | **0xfe** |  |
| **MYSQL\_TYPE\_GEOMETRY** | **0xff** |  |

**ColumnDefinition41**

lenenc\_str catalog

lenenc\_str schema

lenenc\_str table

lenenc\_str org\_table

lenenc\_str name

lenenc\_str org\_name

lenenc\_int length of fixed-length fields [0c]

2 character set

4 column length

1 type

2 flags

1 decimals

2 filler [00] [00]

if command was COM\_FIELD\_LIST {

lenenc\_int length of default-values

string[$len] default values

}

bool Protocol::send\_result\_set\_metadata(List<Item> \*list, uint flags)

Fields

catalog (lenenc\_str) -- catalog (always "def")

schema (lenenc\_str) -- schema-name

table (lenenc\_str) -- virtual table-name

org\_table (lenenc\_str) -- physical table-name

name (lenenc\_str) -- virtual column name

org\_name (lenenc\_str) -- physical column name

next\_length (lenenc\_int) -- length of the following fields (always 0x0c)

character\_set (2) -- is the column character set and is defined in Protocol::CharacterSet.

column\_length (4) -- maximum length of the field

column\_type (1) -- type of the column as defined in Column Type

flags (2) -- flags

decimals (1) -- max shown decimal digits \* 0x00 for integers and static strings \* 0x1f for dynamic strings, double, float \* 0x00 - 0x51 for decimals

## LOCAL INFILE request

## COM\_FIELD\_LIST

## COM\_FIELD\_LIST response

## COM\_CREATE\_DB

## COM\_DROP\_DB

MySQL Server

dispatch\_command

Mysql-5.5

sql/sql\_parse.cc:869

mysql\_binlog\_send

Mysql-5.5

sql/sql\_repl.cc:436

Mysql-5.6

sql/rpl\_master.cc

mysql\_stmt\_get\_longdata

Mysql-5.5

sql/sql\_prepare.cc:2802

mysqld\_stmt\_prepare

Mysql-5.5

sql/sql\_prepare.cc:2165

send\_prep\_stmt

Mysql-5.5

sql/sql\_prepare.cc:328

mysqld\_stmt\_execute

COM\_STMT\_EXECUTE handler: execute a previously prepared statement

Mysql-5.5

/sql/sql\_prepare.cc:2525

mysqld\_stmt\_fetch

COM\_STMT\_FETCH handler: fetches requested amount of rows from cursor

Mysql-5.5

/sql/sql\_prepare.cc:2626

mysqld\_stmt\_close

Delete a prepared statement from memory.

Mysql-5.5

/sql/sql\_prepare.cc:2732

mysqld\_stmt\_reset

Reset a prepared statement in case there was a recoverable error.

Mysql-5.5

/sql/sql\_prepare.cc:2688

Protocol::send\_result\_set\_metadata

Mysql-5.5

/sql/protocol.cc

parse\_com\_change\_user\_packet

Mysql-5.5

/sql/sql\_acl.cc:8253

send\_server\_handshake\_packet

Mysql-5.5

/sql/sql\_acl.cc:7956

Binary Protocol Type Implementation

Protocol\_binary::store\_timeMYSQL\_TIME\*

Mysql-5.5

/sql/protocol.cc:1417

Protocol\_binary::storeMYSQL\_TIME\*

Mysql-5.5

/sql/protocol.cc:1417

Protocol\_binary::store\_short

Mysql-5.5

/sql/protocol.cc:

Field\_year::send\_binary

Mysql-5.5

/sql/field.cc: