public class ServerPrepareHandler implements FrontendPrepareHandler {

private static final Logger LOGGER = LoggerFactory.getLogger(ServerPrepareHandler.class);

private static Escaper varcharEscaper = null;

static {

Builder escapeBuilder = Escapers.builder();

escapeBuilder.addEscape('\'', "\\'");

escapeBuilder.addEscape('$', "\\$");

varcharEscaper = escapeBuilder.build();

}

private ServerConnection source;

private volatile long pstmtId;

private Map<String, PreparedStatement> pstmtForSql;

private Map<Long, PreparedStatement> pstmtForId;

public ServerPrepareHandler(ServerConnection source) {

this.source = source;

this.pstmtId = 0L;

this.pstmtForSql = new HashMap<String, PreparedStatement>();

this.pstmtForId = new HashMap<Long, PreparedStatement>();

}

public void prepare(String sql)

public void sendLongData(byte[] data)

public void reset(byte[] data)

public void execute(byte[] data)

public void close(byte[] data)

public void clear()

// TODO 获取预处理语句中column的个数

private int getColumnCount(String sql)

// 获取预处理sql中预处理参数个数

private int getParamCount(String sql)

// 组装sql语句,替换动态参数为实际参数值

private String prepareStmtBindValue(PreparedStatement pstmt, BindValue[] bindValues)