public class ServerConnection extends FrontendConnection {

protected static final long AUTH\_TIMEOUT = 15 \* 1000L;

protected volatile int txIsolation;

protected volatile boolean autocommit;

protected volatile boolean preAcStates; // 上一个ac状态,默认为true

protected volatile boolean txInterrupted;

protected volatile String txInterrputMsg = "";

protected long lastInsertId;

protected NonBlockingSession session;

protected volatile boolean isLocked = false;

public ServerConnection(NetworkChannel channel) throws IOException {

super(channel);

this.txInterrupted = false;

this.autocommit = true;

this.preAcStates = true;

}

public ServerConnection() throws IOException {

super();

this.txInterrupted = false;

this.autocommit = true;

this.preAcStates = true;

}

public boolean isIdleTimeout()

public int getTxIsolation()

public void setTxIsolation(int txIsolation)

public boolean isAutocommit() {

public void setAutocommit(boolean autocommit)

public long getLastInsertId()

public void setLastInsertId(long lastInsertId)

public void setTxInterrupt(String txInterrputMsg)

public boolean isTxInterrupted()

public NonBlockingSession getSession2()

public void setSession2(NonBlockingSession session2)

public boolean isLocked()

public void setLocked(boolean isLocked)

public void ping()

@Override

public void heartbeat(byte[] data)

public void execute(String sql, int type)

private boolean isNormalSql(int type)

public RouteResultset routeSQL(String sql, int type)

public void routeEndExecuteSQL(String sql, final int type, final SchemaConfig schema)

public void commit()

public void rollback()

public void lockTable(String sql)

public void unLockTable(String sql)

public void cancel(final FrontendConnection sponsor)

public void close(String reason)

public static int appearNumber(String srcText, String findText)

public String toString()

public boolean isPreAcStates()

public void setPreAcStates(boolean preAcStates)

}