

exchange Orderme detailed design specificatioN

Exchange OrderME v0.1

Savin 2018-01-22

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Version** | **Revised by** | **Time** | **Remarks** |
| 1 | v0.1 | Savin | 2018-01-22 | 初版 |
|  |  |  |  |  |
|  |  |  |  |  |

# CONTENT

[CONTENT 3](#_Toc504466136)

[1. 位置 3](#_Toc504466137)

[2. 服务时序图设计 4](#_Toc504466138)

[3. 服务设计 5](#_Toc504466139)

[4. 处理流程设计 6](#_Toc504466140)

[5. 消息设计 7](#_Toc504466141)

[5.1 内部服务之间消息： 7](#_Toc504466142)

[5.1.1订单委托 :OrderME 🡨-- OrderSrv 8](#_Toc504466143)

[5.1.2订单委托 :OrderME --🡪 OrderSrv 8](#_Toc504466144)

[5.2本服务和ME系统的消息设计 9](#_Toc504466145)

[6. 数据库表设计 9](#_Toc504466146)

[7. 重要类设计 9](#_Toc504466147)

# 位置



OrderME 是Exchange System 和ME system的订单方面的交互桥梁

* 登录
* 下单
* 撤单
* 订单状态查询

# 服务时序图设计



# 服务设计

2.1 内部设计



* 1. 要点说明

Pusher Thread 可以把消息推送给原发送者(比如说：OrderSrv5 发出的下单消息，ME回复OrderME可以把该消息转发给OrderSrv5)

# 处理流程设计

<Todo: 服务内部某模块的流程图+简要说明,可多个>

类似下图:

# 消息设计

<Todo: 本服务和其他服务交互的消息,可多个>

原则：内部消息字节序，外部消息JSON,FIX

## 5.1 内部服务之间消息：

### 5.1.1订单委托 : OrderSrv 🡪 OrderME

|  |
| --- |
| typedef struct \_\_STOrder  {  int Tag; //0x1000  unsigned int AccoutID; //交易所商户的ID,ME分配的  unsigned long OrderID //订单ID  float Price; //交易价格，这里最多精确到小数点后6位  float OrderNumber ; //订单数量  char OrderType ; //订单类型，0 表示限价订单，1表示市场订单  char TransType ; //交易类型，0 表示买， 1表示卖  char Symbol[22]; //交易品种(字符串类型)，  //用于标识是哪一种交易，比如BTC/LTC  } STOrder; |

### 5.1.2订单委托ACK : OrderME 🡨-- OrderSrv

|  |
| --- |
| typedef struct \_\_STOrderAck  {  int Tag; //0x1001  unsigned long OrderIndex; //ME分配的唯一订单映射编号。  unsigned long OrderID; //提交时的订单编号  char OrderStatus; //订单状态，0：接收, 1:拒绝  char TransactTime[31]; //由撮合中心发回的接收该订单时间  char Text[128]; //如果是订单被拒绝，在这里则填入拒绝的原因  } STOrderAck; |

### 5.1.3 撤单请求 : OrderSrv 🡪 OrderME

|  |
| --- |
| typedef struct \_\_STCanelOrder  {  int Tag; //0x1002  unsigned long OrderIndex; //ME分配的唯一订单映射编号。  unsigned long OrderID; //提交时的订单编号  float Price ; //交易价格，这里最多精确到小数点后6位  char TransType; //交易类型，0 表示买， 1表示卖  char Symbol[23]; //交易品种(字符串类型)，  } STCanelOrder; |

### 5.1.4 撤单请求ACK : OrderME 🡪 OrderSrv

|  |
| --- |
| typedef struct \_\_STCanelOrderAck  {  int Tag; //0x1003  unsigned long OrderID; //提交时的订单编号  char OrderStatus; //订单状态，0：接收, 1:拒绝  char TransactTime[32]; //由撮合中心发回的接收该订单时间  char Text[128]; //如果是订单被拒绝，在这里则填入拒绝的原因  } STCanelOrderAck; |

### 5.1.5 订单状态查询 OrderSrv 🡪 OrderME

|  |
| --- |
| typedef struct \_\_STQueryOrderStatus  {  int Tag; //0x1004  unsigned int AccoutID; //交易所商户的ID,ME分配的  unsigned long OrderID; //提交时的订单编号  } STQueryOrderStatus; |

### 5.1.6订单状态查询返回 : OrderME 🡪 OrderSrv

|  |
| --- |
| typedef struct \_\_STQueryOrderStatusReturn  {  int Tag; //0x1005  unsigned long OrderID；//订单编号  char OrderStatus //订单状态， A = 挂单中 1 = 完成成交 2 = 部分成交，  //3 = 部分撤销 4 = 已撤销 6 = 撤单处理中  Char Type //交易类型，0 表示买， 1表示卖  Char Symbol[24] //交易品种(字符串类型)，用于标识是哪一种交易，  //比如BTC/LTC  Float OrderPrice ; //原始订单价格，这里最多精确到小数点后6位  float OrderNum ; //原始订单数量  float AvgPx ; //平均执行价格  float ExcNum ; //总执行数量  char TransactTime[32] ;由撮合中心发送的时间  } STQueryOrderStatusReturn; |

## 5.2本服务和ME系统的消息设计

采用FIX4.4 协议

详见ME API文档

# 数据库表设计

<Todo: 本服务涉及到的数据库表,可多个>

# 重要类设计