**服务器端第一阶段接口需求文档**

1. 系统架构
2. 数据库设计

使用MongoDB，提高数据库查询速度

通过第三方接口接收服务器信息，并保存至本地数据库

通过websocket协议，为用户提供查询服务

1. 用户查询接口
2. 请求

{

"data" : ["type","kType","exCode1","conCode1","exCode2","conCode2", …]

}

**请求类型type:**

HET：心跳包, 暂时不做处理

CON: 查询合约

MAR: 查询行情

KLE: 查询K线

**K线类型kType:**

"ONE","THR","FIV","TEN","HAF","SIT","FOH","DAY";

分别代表1 3 5 10 30 60 4hours 1day

**交易所代码exCode：**

例如“CME”

“ALL”表示所有交易所

**合约代码conCode：**

例如“CL1712”

“ALL”表示所有合约

1. 响应
2. 合约响应

{

"state" : 1, //1表示请求成功，0表示请求失败

"data" : {

"交易市场" : {

"交易品种" : ["品种合约", "品种合约", …],

"交易品种" : ["品种合约", "品种合约", …],

……

},

"交易市场" : {

"交易品种" : ["品种合约", "品种合约", …],

"交易品种" : ["品种合约", "品种合约", …],

……

},

……

}

}

1. 行情响应

{

"state" : 1, //1表示请求成功，0表示请求失败

"data" : [

[交易日,合约代码,交易所代码,合约在交易所的代码,最新价,现手,持仓量,上次结算价,本次结算价,涨跌,涨跌幅,当日均价,买量,卖量,申买价一,申买量一,申买价二,申买量二,申买价三,申买量三,申买价四,申买量四,申买价五,申买量五,申卖价一,申卖量一,申卖价二,申卖量二,申卖价三,申卖量三,申卖价四,申卖量四,申卖价五,申卖量五],

[交易日,合约代码,交易所代码,合约在交易所的代码,最新价,现手,持仓量,上次结算价,本次结算价,涨跌,涨跌幅,当日均价,买量,卖量,申买价一,申买量一,申买价二,申买量二,申买价三,申买量三,申买价四,申买量四,申买价五,申买量五,申卖价一,申卖量一,申卖价二,申卖量二,申卖价三,申卖量三,申卖价四,申卖量四,申卖价五,申卖量五],

……

]

}

1. K线响应

{

"state" : 1, //1表示请求成功，0表示请求失败

"data" : [ //300个点

[最大值,最小值,涨幅,跌幅,…]

……

]

}