

UI设计测试题

概述：

设计一个为证券交易市场显示下单列表和成交列表的Web UI。

一个单(Order)由（单号(number)，买(bid)/卖(ask)，数量(quantity)，价格(price)）表示，例如

- (1001, ask, 6, 480)
- (1002, bid, 8, 470)
- (1003, bid, 5, 460)

当市场上出现一个买单的价格高于一个卖单的时候，两单（互为对手单）可以成交，成交价格为两单价格的平均数。当一个单可以和多个对手单成交时，优先和价格更优的先成交；价格相同的优先和编号更小的成交。

比如上面的例子，最低的卖单价(480)比最高的买单价(470)还要高，所以无成交。若此时新报一单（1004, ask, 10, 460），则该单先与1002成交，成交价为465，成交量为 $\min(10, 8) = 8$ ；而后再与1003成交，成交价为460，成交量为 $\min(10 - 8, 5) = 2$ 。之后Order book变为

- (1001, ask, 6, 480)
- (1003, bid, 3, 460)

后端API：

后端每1到2秒左右生成一个新的order，order number从0开始连续递增；价格均为整数。后端为前端在TCP 9888端口提供两个http API：

1. GET "/listOrders?start=<start order number>&size=<how many orders to fetch>"

例如：<http://0.0.0.0:9888/listOrders?start=10&size=100>

获取从编号为10的order开始的100个order。若当前最后一个order编号 $n < 110$ ，则返回编号从10到 n 的order；若 $n < 10$ ，则返回空。

返回结果示例：

```
[{"number":10,"side":"ask","quantity":13,"price":160}, {"number":11,"side":"ask","quantity":10,"price":87}]
```

2. GET "/reset"

重置后端状态：清空所有order；新order从0开始重新编号。该命令主要用于调试和演示。

前端设计需求：

- UI展示两个队列：
 - 下单队列，该队列展示尚未成交的价格最优的买单和卖单各20个，按照价格从高到低排序，同价格按照单号从小到大排序。

- 成交队列，显示最近30笔成交，按照成交的时间由新到旧排序。点击成交队列中的item，可展开显示成交信息，包括成交时间，成交价格，对手买卖单信息等。动态效果：加入新的成交单时，现有队列中的元素向下移动，腾出位置。

代码下载与提交：

1. Fork https://github.com/libreoscar/ethmarket_interview
2. 配置go 1.6环境（go语言的安装和配置见[这里](#)）。
3. `go get github.com/libreoscar/ethmarket_interview`
4. 在ethmarket_interview目录下运行`go run main.go`
5. 在<http://0.0.0.0:9888/>可以看到一个向后端获取数据并显示的示例，对应的js文件在static/目录下。

请按要求开发前端，向9888端口的server获取所需数据，并生成UI。ethmarket_interview中的示例UI使用了Vue.js，但并不要求测试者使用同样的技术。开发完成后请将代码上传至你的github repo以供查看。