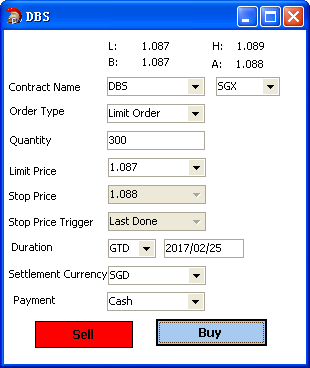
WeChat设计文档

1. Trading
   1. UI design



1. L: low H: high B:Bid A:ask
2. Contract Name暂时自己只填入从watchlist传过来的合约名，后面是Exchange，暂时也自己填相应合约的exchange
3. Order type：订单类型combobox，只能选择，controller会提供列表
4. Quantity: input box，只能输入正整数
5. Limit Price: combo box，可以选择，可以输入，controller会提供列表，用户输入的数值必须在列表中
6. Stop Price：combo box，可以选择，可以输入，controller会提供列表，用户输入的数值必须在列表中，该控件只在Order Type选择为Stop limit order时才enable
7. Stop Price Trigger: combo box，只可选择，controller会提供列表，该控件只在Order Type选择为Stop limit order时才enable
8. Duration：combo box，只可选择，controller会提供列表，后面是一个日期控件。当选择Day时，后面的日期填入当天日期，不可设置；当选择为GTD时，可选择一个有效日期（大于当前日期，且小于最长天数，最大天数controller会提供）
9. Settlement Currency: combo box，只可选择，controller会提供列表
10. Payment: combo box，只可选择，controller会提供列表
    1. Flow
11. 首先调用StocksController.GetStockInfo(string Market, string CompanyCode),如果返回code＝1，根据返回值填充UI。如果返回code不为1，直接显示msg。
12. 用户对UI进行操作，比如设置价格，数量等等，自行进行校验。比如当焦点离开Limit Price时，对输入的价格进行校验，检验失败，则光标定位于Limit Price。
13. 用户点击Buy或Sell按钮后，调用StocksController.ValidateAdvancedOrder进行校验。

If(code=-1)

弹出只有一个OK按钮的消息框，显示msg。用户点击OK后返回Trading界面。

If(code=-42)

弹出只有一个OK按钮的消息框，显示msg。用户点击OK后退出程序（需重新登录）。

If(code=-333)

弹出有Yes, No按钮的消息框,显示msg，询问用户是否继续下单。如果用户点击No，返回Trading界面，如果用户点击Yes,进入4)。

If(code=1)，代表校验成功，直接进入4)。

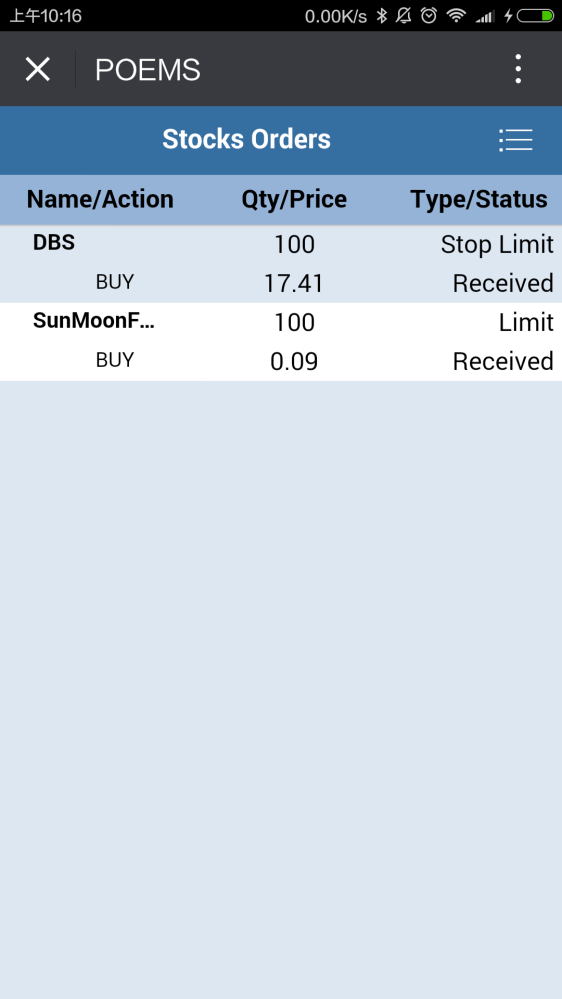
1. 显示确认界面，有Cancle, Submit两个按钮。将用户设置的相关数据进行显示，显示密码输入框，提示用户输入下单密码。

如果用户点击Cancle，返回Trading界面。

如果用户点击Submit，校验密码是否正确。如果正确，进入下一步。

1. 调用StocksController.SubmitAdvancedOrder，弹出只有一个OK按钮的消息框，显示msg。如果code!=1,用户点击OK后，留在Trading界面。否则切换到OrderStatus界面，这样用户就可以看到刚才下的定单。
2. Withdraw/Amend Order

讨论：Withdraw/Amend 需要在Orders页面操作，那么UI如何设计？



基础知识：

1. 并不是所有的订单都可withdraw或者amend。当前的页面(getTodayOrders)还没有取得withdraw/amend状态，getOrderDetails可以返回这些状态。
2. Amend order时只能修改Quantity。

备选方案：

1. 选中一行向左滑动，弹出2个按钮，一个withdraw，一个amend。点击withdraw直接调用StocksController.WithdrawOrder(OrderNo, Market);点击amend，弹出一个New Quantity输入框，让输入新数量，然后点击确认，直接调用StocksController.AmendOrder(OrderNo, Market, NewQuantity);

类似效果如下：

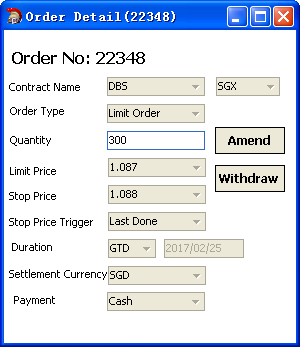


1. 长按一行，弹出pop up菜单，两个menu item: withdraw, amend。点击结果同1)。



1. 点击一行，弹出order detail页面，Order detail页面可以重用Trading页面。

效果如下：



2.1 UI设计

2.2 Flow

1) 首先调用StocksController.GetOrderDetail，获取信息，并填充UI。

Amendable： Y：可修改；N:不可修改

Withdrawable：Y：可撤销；N:不可撤销

数量修改范围：[LotSize, AmendableQty)

比如LotSize为100， AmendableQty为1000时，用户只能输入以下数量：100, 200, 300,......900

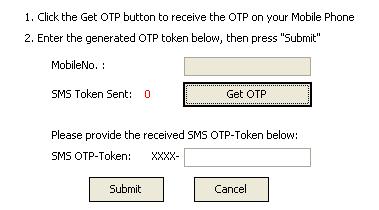
2) 撤销订单：StocksController.WithdrawOrder(OrderNo, Market)

修改订单：StocksController.AmendOrder(OrderNo, Market, NewQuantity)

1. 2FA

用户登录成功后，判断Modules\V1\_Authenticate.Authenticate方法中返回的TwoFAEnable，如果该值为1，则需要进行2FA验证。弹出2FA验证页面。提供2种验证方式SMS和Token供用户选择，两种方式说明如下。

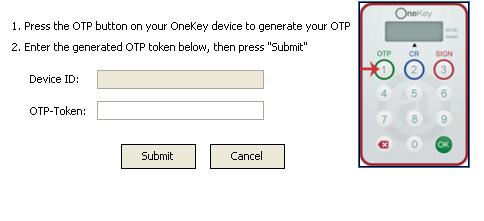
* 1. SMS



调用GetClientTwoFAInfo，取得DeviceType为0103对应Device的MobileNo，填入UI。同时可取得NAFUserName, DeviceID。

当用户点击Get OTP按钮时，调用GetOutOfBondOTP(accountNo, NAFUserName, DeviceID, DeviceType, sessionId, SPID, TwoFAAppID)。SPID和TwoFAAppID在调用Authenticate方法时可得到。如果成功，用户的手机会收到验证码。用户将验证码输入SMS OTP-Token。当用户点击Submit按钮后，调用AuthenticateTwoFAUser(accountNo, NAFUserName, DeviceID, DeviceType, "xxxxxxxxx", Challenge, sessionId, SPID, TwoFAAppID);其中参数Challenge会在GetOutOfBondOTP返回。"xxxxxxxxx"为用户输入的验证码。如果验证成功，则登录成功，进入主界面。如果验证失败，显示失败消息，还停留在该页面。

* 1. Token



调用GetClientTwoFAInfo，取得DeviceType为0101对应Device的DeviceID，填入UI。同时可取得NAFUserName。

用户输入OTP-Token后点击Submit，调用AuthenticateTwoFAUser(accountNo, NAFUserName, DeviceID, DeviceType, "xxxxxxxxx", "", sessionId, SPID, TwoFAAppID); SPID和TwoFAAppID在调用Authenticate方法时可得到。"xxxxxxxxx"为用户输入的OTP-Token。如果验证成功，则登录成功，进入主界面。如果验证失败，显示失败消息，还停留在该页面。

1. TODO