**行情分析小程序**

关于本行情分析小程序，主要用于行情数据的详细数据显示、分析以及数据图表显示。基本的设计、数据以及预测数据走势等依照股票分析软件来设计。

设计主要按照以下几个步骤进行。

*  [初步页面设计](https://github.com/guanggong2020/quotation#%E5%88%9D%E6%AD%A5%E9%A1%B5%E9%9D%A2%E8%AE%BE%E8%AE%A1)
*  [静态页面编写](https://github.com/guanggong2020/quotation#%E9%9D%99%E6%80%81%E9%A1%B5%E9%9D%A2%E7%BC%96%E5%86%99)
*  [动态数据及传参](https://github.com/guanggong2020/quotation#%E5%8A%A8%E6%80%81%E6%95%B0%E6%8D%AE%E5%8F%8A%E4%BC%A0%E5%8F%82)
*  [图表设计及代码](https://github.com/guanggong2020/quotation#%E5%9B%BE%E8%A1%A8%E8%AE%BE%E8%AE%A1%E5%8F%8A%E4%BB%A3%E7%A0%81)
*  [函数、功能等设计](https://github.com/guanggong2020/quotation#%E5%87%BD%E6%95%B0%E5%8A%9F%E8%83%BD%E7%AD%89%E8%AE%BE%E8%AE%A1)
*  [数据库数据获取api](https://github.com/guanggong2020/quotation#%E6%95%B0%E6%8D%AE%E5%BA%93%E6%95%B0%E6%8D%AE%E8%8E%B7%E5%8F%96api)
*  [后续功能修改或增加](https://github.com/guanggong2020/quotation#%E5%90%8E%E7%BB%AD%E5%8A%9F%E8%83%BD%E4%BF%AE%E6%94%B9%E6%88%96%E5%A2%9E%E5%8A%A0)

目前已完成了前4部分，后续会根据需要做相应的调整、修改。

**初步页面设计**

参考东方财富、智与良投等行情软件，加上自己的设计思路，初步界面设计如下图:

[](https://github.com/guanggong2020/quotation/blob/master/imgs/pre_design.png)

初步设计方案中，包括了主页（index）、详细数据页（details）和搜索页（seach）三页。每页中都有搜索图标、点击输入搜索内容即可跳转或更新到搜索页面。主页主要是预览和排序、比较各支股票等大概信息；详细数据页包含了该股票的各项信息！（具体由所获得的数据做具体调整）、走势图、预测图等；搜索数据页与主页相差无几。

**目前设计图样如下:**

[](https://github.com/guanggong2020/quotation/blob/master/imgs/index.jpg) [](https://github.com/guanggong2020/quotation/blob/master/imgs/details.jpg) [](https://github.com/guanggong2020/quotation/blob/master/imgs/seaching.jpg) [](https://github.com/guanggong2020/quotation/blob/master/imgs/seach.jpg)

**静态页面编写**

根据设计方案，先编写静态页面。主要是编写wxml、wxss两种文件，一个是负责页面布局、一个是负责页面样式。目前为止，此部分基本完成。主页（index）、详细数据页（details）和搜索页（seach）三页的静态样式和页面都已全部完成在对应的文件夹下，此三页都已进入下一步骤。

**动态数据及传参**

根据编写完成的静态页面做相应的调整，主要是页面跳转的传值问题和动态数据和布局问题。

关于传参问题，目前采用通用的地址附带参数的方式，即：url?values=数据内容。因为预设采用传股票的代码信息，再利用该信息查询数据库的方式，所以这种方式相对简单实用。

关于动态数据问题，目前自定义一套数据定义，具体如下：

/\*\*

\* code:代码

\* name:名称

\* current:当前价

\* pre\_close:昨收价

\* change\_pct:涨幅 (当前-昨收)/昨收

\* change:涨跌 (当前-昨收)

\* open:今开盘

\* volume:成交量

\* turnover\_ratio:换手率

\* high:最高

\* low:最低

\* sum\_amount:成交额 (买入成交额+卖出成交额)

\* sell:内盘

\* buy:外盘

\* market\_cap:总市值(亿元)

\* pe:市盈率

\* amplitude:振幅 （当日最高点的价格－当日最低点的价格）/前收价

\* circulating\_market\_cap:流通市值

\*/

采用上述定义来对静态页面修改数据。同时利用微信的wx:for、wx:if等函数实现动态数据加载和动态样式修改。

**图表设计及代码**

目前图表的设计和代码使用的是图表插件wxcharts.js，代码来源：[https://github.com/xiaolin3303/wx-charts/tree/master/dist。](https://github.com/xiaolin3303/wx-charts/tree/master/dist%E3%80%82)

后续会考虑自编图表代码或者优化、修该已有插件代码实现更为优化图表。

**函数、功能等设计**

目前该部分还尚未达到，**主要预设的功能有：**

* 自选股票等相关内容（包含登录）
* 数据比较
* 数据动态预测
* 实时同步

**预设函数部分有:**

* 数据排序
* 登录获取用户相关信息
* 数据预测

**数据库数据获取api**

此部分是编写数据库数据获取的主要部分，接下来主要测试这一部分。

**后续功能修改或增加**

根据需求和现实情况做动态调整，目前预计的增添、修改如下：

*  个人信息页面
*  数据比较页面

**参考文档**

* [云开发文档](https://developers.weixin.qq.com/miniprogram/dev/wxcloud/basis/getting-started.html)
* [小程序组件开发文档](https://developers.weixin.qq.com/miniprogram/dev/component/)
* [小程序API开发文档](https://developers.weixin.qq.com/miniprogram/dev/api/)
* [wxcharts.js使用文档](https://github.com/xiaolin3303/wx-charts/issues/58)