详细设计计划

—SE2018—G04简易查

**小组：G04**

**骆佳俊 徐双铅 吕迪**

**2018.6.24**

**修订表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **生成版本** | **修订人** | **修订章节与内容** | **修订日期** |
| 1 | 0.5 | 吕迪 | 编写内容 | 2018.5.6 |
| 2 | 0.6 | 吕迪 | 添加详细实现计划 | 2018.5.7 |
| 3 | 0.7 | 吕迪 | 增加PAD图 | 2018.5.14 |
| 4 | 0.8 | 吕迪 | 增加逻辑图 | 2018.5.14 |
| 5 | 0.9 | 吕迪 | 修改推送方式 | 2018.5.18 |
| 6 | 1.0 | 吕迪 | 具体化预警设置修改PAD图 | 2018.6.03 |
| 7 | 1.2 | 吕迪 | 微信推送条件具体化 | 2016.6.03 |
| 8 | 1.0 | 吕迪 | 根据需求增加个股实时走势图 | 2018.6.03 |
| 9 | 1.2 | 吕迪 | 根据需求预警时在实时走势图显示预警灯泡 | 2016.6.18 |
| 10 | 1.3 | 吕迪 | 增加实时走势图的PAD图 | 2016.6.18 |
| 11 | 1.4 | 吕迪 | 增加格式转换PAD图 | 2016.6.19 |
| 12 | 1.5 | 吕迪 | 推送HIPO图更改 | 2016.6.20 |
| 13 | 1.6 | 骆佳俊 | 增加了界面设计和算法设计 | 2016.6.24 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目录**

[1 引言 5](#_Toc517640556)

[1.1 编写目的 5](#_Toc517640557)

[1.2 背景 5](#_Toc517640558)

[1.3 定义 5](#_Toc517640559)

[1.4 参考资料 6](#_Toc517640560)

[2 系统结构 6](#_Toc517640561)

[2.1 系统结构框图 6](#_Toc517640562)

[2.2 HIPO图 7](#_Toc517640563)

[2.3 IPO表 8](#_Toc517640564)

[3 建库模块设计说明 10](#_Toc517640565)

[3.1 模块描述 10](#_Toc517640566)

[4 输入模块设计说明 11](#_Toc517640567)

[4.1 模块描述 11](#_Toc517640568)

[4.2 功能 11](#_Toc517640569)

[4.3 性能 11](#_Toc517640570)

[4.4 输入项 11](#_Toc517640571)

[4.5 输出项 11](#_Toc517640572)

[4.6 流程逻辑 11](#_Toc517640573)

[4.6.1 输入股票代码 12](#_Toc517640574)

[4.6.2 输入自定义条件 13](#_Toc517640575)

[4.7 存储分配 15](#_Toc517640576)

[4.8 限制条件 15](#_Toc517640577)

[4.9 测试计划 15](#_Toc517640578)

[5 匹配模块设计说明 15](#_Toc517640579)

[5.1 模块描述 15](#_Toc517640580)

[5.2 功能 15](#_Toc517640581)

[5.3 性能 16](#_Toc517640582)

[5.4 输入项 16](#_Toc517640583)

[5.5 输出项 16](#_Toc517640584)

[5.6 设计方法（算法） 16](#_Toc517640585)

[5.7 流程逻辑 16](#_Toc517640586)

[5.8 存储分配 17](#_Toc517640587)

[5.9 限制条件 17](#_Toc517640588)

[5.10 测试计划 17](#_Toc517640589)

[6 输出模块设计说明 18](#_Toc517640590)

[6.1 模块描述 18](#_Toc517640591)

[6.2 功能 18](#_Toc517640592)

[6.3 性能 18](#_Toc517640593)

[6.4 输入项 18](#_Toc517640594)

[6.5 输出项 18](#_Toc517640595)

[6.6 设计方法（算法） 18](#_Toc517640596)

[6.7 流程逻辑 19](#_Toc517640597)

[6.7.1 股票咨询显示 19](#_Toc517640598)

[6.7.2 格式转换 20](#_Toc517640599)

[6.7.3 微信预警推送 21](#_Toc517640600)

[6.7.4 个股实时走势图 22](#_Toc517640601)

[6.7.5 预警灯泡 23](#_Toc517640602)

[6.8 存储分配 23](#_Toc517640603)

[6.9 限制条件 23](#_Toc517640604)

[6.10 测试计划 23](#_Toc517640605)

[7 详细的实现计划 24](#_Toc517640606)

[7.1 模块分类 24](#_Toc517640607)

[7.2 代码设计分工 24](#_Toc517640608)

[7.3 代码实现分工 24](#_Toc517640609)

[7.4 代码测试分工 24](#_Toc517640610)

# 引言

伴随着资金的增长和对富有的向往，越来越多的人开始加入炒股的行列。股市的行情是瞬息多变的，掌握的实时的资料是十分必要的，我们做一个可以查看股价，并提供监控提醒。方便股民在工作空余时间实时查看股价，在股民事务繁忙情况下提供监控实时提醒。股民根据信息作出判断。

## 编写目的

编写详细设计说明书的目的是为了使开发人员在完成概要设计说明书的基础上，完成概要设计的各项功能规定；为程序员写出实际的程序代码提供依据。它是软件设计阶段所有任务和所有相关人员所需的参考资料。预期的读者为系统的开发者、使用者和指导老师。

## 背景

1. 待开发软件的名称为“简易看”；
2. 本项目提出者和开发者均为SE2018-春-G04小组，目标人群设定为炒股和偶尔想要了解股价的人群；
3. 相互来往关系：本软件通过调用阿里API，搜索相应股票的信息，根据对返回json数据解析，告诉用户相应信息。

## 定义

API：Application Programming Interface,应用程序编程接口

## 参考资料

1. 张海藩,牟永敏.软件工程导论（第六版）[M].北京：清华大学出版社, 2013:117-136
2. ISO标准软件工程文档模板 ：百度文库 （2018.5.6）

<https://wenku.baidu.com/view/b06aab6327d3240c8447ef23.html>

1. 详细设计报告 ：豆丁网（2018.5.6）

<http://www.docin.com/p-793962191.html>

1. 总结报告：详细设计说明书实例：百度文库（2018.5.6）

<https://wenku.baidu.com/view/ab2bfcd1ad02de80d4d840e7.html>

1. pdl语言：百度百科（2018.5.6）

[https://baike.baidu.com/item/pdl%E8%AF%AD%E8%A8%80/8411985?fr=aladdin](https://wenku.baidu.com/view/ab2bfcd1ad02de80d4d840e7.html)

1. 软件工程—软件工程用图 ：博客园（2018.5.6）

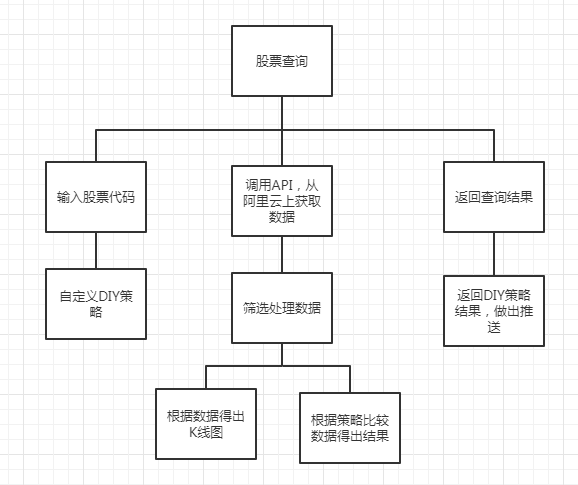
<https://www.cnblogs.com/kzloser/archive/2012/07/05/2577432.html>

1. 注释设计：百度知道（2018.5.6）

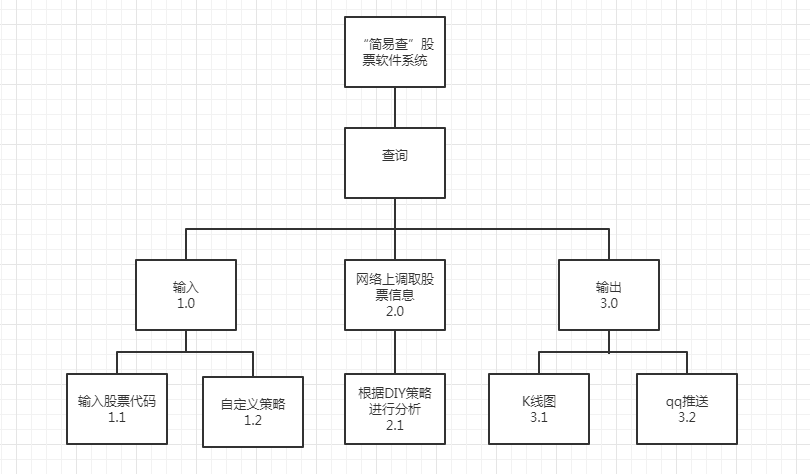
<https://zhidao.baidu.com/question/1671772153438404347.html>

# 系统结构

## 系统结构框图



## HIPO图



## IPO表

**输入模块IPO表**



**处理模块IPO表：**



**输出模块IPO表**



# 建库模块设计说明

## 模块描述

无

# 输入模块设计说明

## 模块描述

本模块读入用户输入的股票代码，自定义股票预警信息，生成正确的API，使得系统能正常的进行咨询。

## 功能

读取用户输入的股票代码，将代码进行读取和编译，进行API咨询。

读取用户输入的自定义股票预警信息，放入后台进行实时监控。

## 性能

灵活性：本系统所调用的阿里云检索API为“股票咨询API”；

时间特性：咨询时间<10秒。

## 输入项

股票代码，符合标准的股票代码。

自定义预警信息。

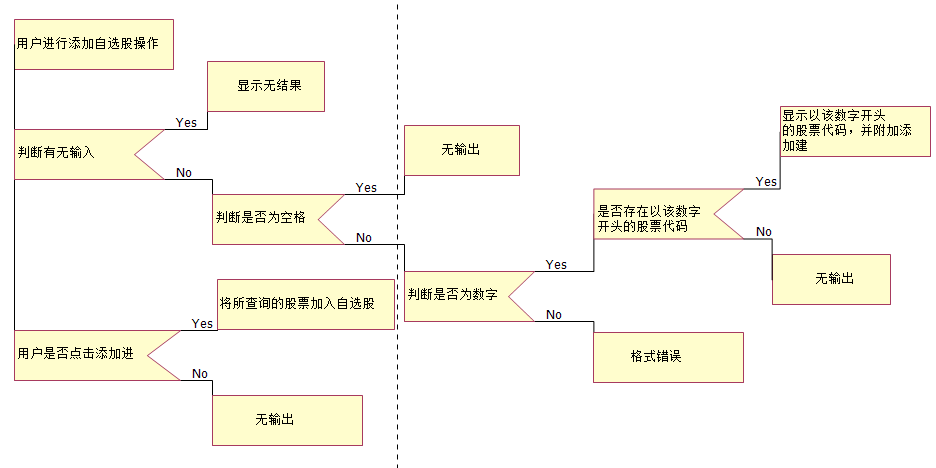
## 输出项

输出项String类型的json数据。

预警方面的输出项为微信预警提示。

## 流程逻辑

### 输入股票代码



PAD图

Procedure 输入股票代码 is

读取用户输入股票代码s

If(无输入)

显示无结果

抛出异常

if（为空）

无输出

抛出异常

if（为数字）

if（存在该数字开头的股票代码）

显示，附加添加键

Else

无输出

Else

输出格式错误

Else

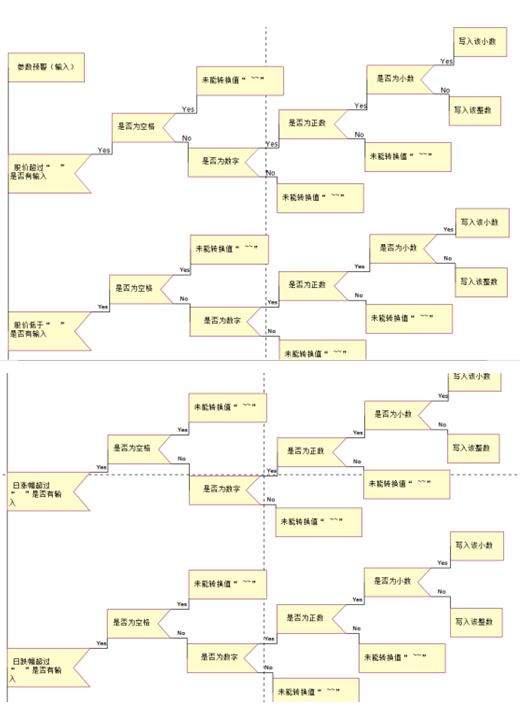
五输出

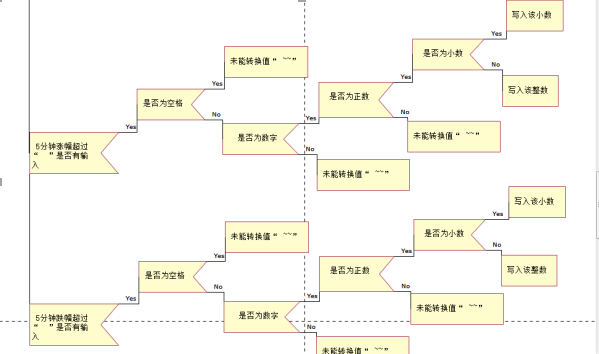
Else

显示无结果

End if

### 输入自定义条件





PAD图

Procedure 自定义设计 is

读入用户自定义规则下输入的值，股票价格是否超过，股票价格是否低于，股票日涨幅是否超过，股票日涨幅是否低于，股票五分钟涨幅是否超过，股票五分钟内跌幅是否低于

If(股票价格是否超过为空)

Null

else if(股票价格是否低于不为数字)

抛出异常

else

设定自定义规则1成功

If(股票价格是否低于为空)

Null

else if(股票价格是否低于不为数字)

抛出异常

else

设定自定义规则2成功

If(股票日涨幅是否超过为空)

Null

else if(股票日涨幅是否超过不为数字)

抛出异常

else

设定自定义规则3成功

If(股票日涨幅是否低于为空)

Null

else if(股票日涨幅是否低于不为数字)

抛出异常

else

设定自定义规则4成功

If(股票五分钟涨幅是否超过为空)

Null

else if(股票五分钟涨幅是否超过不为数字)

抛出异常

else

设定自定义规则5成功

If(股票五分钟内跌幅是否低于为空)

Null

else if(股票五分钟内跌幅是否低于不为数字)

抛出异常

else

设定自定义规则6成功

End if

End 自定义设计

## 存储分配

本程序进行编码所用编程语言为C#，直接内存分配由C#运行时分配。

本组件内所以来的变量、结构要求全部在组件元素内申明。

## 限制条件

1. 输入股票代码需是符合标准的股票代码。
2. 输入的自定义预警信息只能在给定条件中选择。
3. 仅支持windows系统。

## 测试计划

* 技术要求：正常访问获得股票信息，微信正常预警推送。
* 输入数据：符合规则的股票代码和自定义预警信息。
* 预期结果：能正常的咨询到股票信息和微信正常预警。
* 进度安排：10天之内完成。
* 人员职责：完成此模块的编码、测试、调试。
* 设备条件：装有visual studio的电脑。

# 匹配模块设计说明

## 模块描述

本模块完成调用阿里云搜索API模块，得到检索结果

## 功能

将输入股票代码后得到的string数据传入阿里云检索API接口，得到检索结果。

## 性能

灵活性：无

时间特性：视网络响应速度而定

## 输入项

输入模块输出的String类型的代码数据。

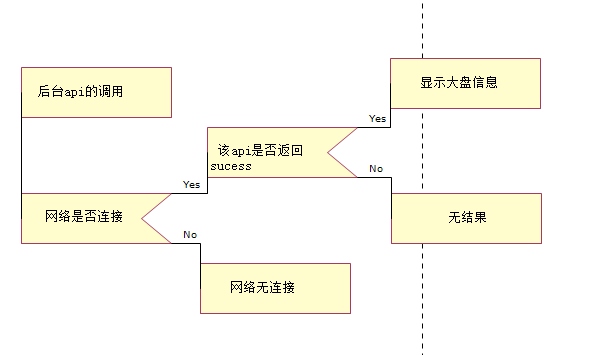
## 输出项

String类型的检索结果。

## 设计方法（算法）

本模块直接调用阿里云检索API接口，没有自定义的算法。

## 流程逻辑



PAD图

Procedure 数据咨询 is

If（网络无连接）

抛出异常

Else if（api返回success）

显示大盘信息

Else

抛出异常

End if

End 数据咨询

## 存储分配

本程序进行编码所用编程语言为C#，直接内存分配由C#运行时分配。

本组件内所以来的变量、结构要求全部在组件元素内申明。

## 限制条件

1. 仅支持windows系统下的电脑。
2. 必须在有网络的条件下进行。

## 测试计划

* 技术要求：成功调用阿里云检索API接口。
* 输入数据：String类型的股票代码数据。
* 预期结果：正确得到检索结果。
* 进度安排：10天。
* 人员职责：完成此模块的编码、测试、调试。
* 设备条件：装有C# 的电脑。

# 输出模块设计说明

## 模块描述

该模块为输出模块，包含检索结果列表、股票的详细信息；该模块的主要任务是整个系统的信息输出；该模块的主要特点是：作为本系统的后一个模块，是所有信息的输出部分，是实现系统功能中的重要模块。

## 功能

获得检索结果列表；

输出相应信息。

微信预警推送。

实时走势图的输出

实时走势图小灯泡展示

## 性能

对符合要求的股票代码，能给与正确的实时股票信息。

## 输入项

检索结果（JSON格式）。

## 输出项

股票的各项信息。

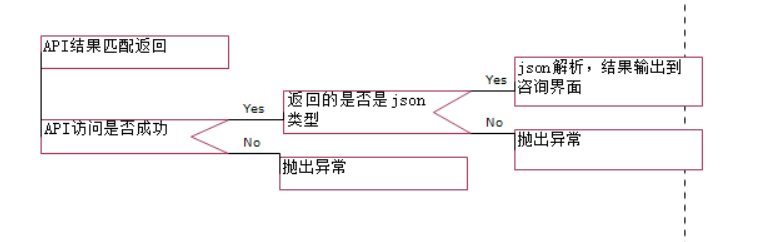
微信预警推送。

## 设计方法（算法）

无

## 流程逻辑

### 股票咨询显示



PAD图

Procedure 股票咨询显示 is

对返回的json解析

If(返回结果失败)

抛出异常

返回主界面

else if(返回结果解析失败)

抛出异常

返回主界面

else

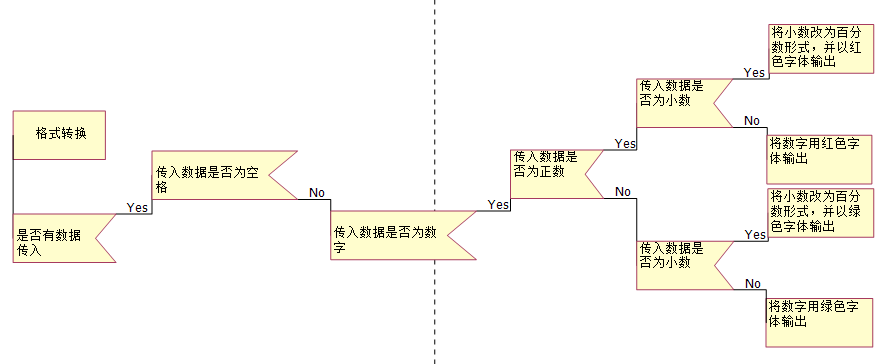
咨询成功

显示该股票各个信息

End if

End股票咨询显示

### 格式转换



If（有数据传入）

If（传入数据不为空）

If（传入数据为数字）

If（传入数据为正数）

If（传入数据为小数）

讲小数改为百分数形式，并以红色字体输出

Else

讲数字用红色字体输出

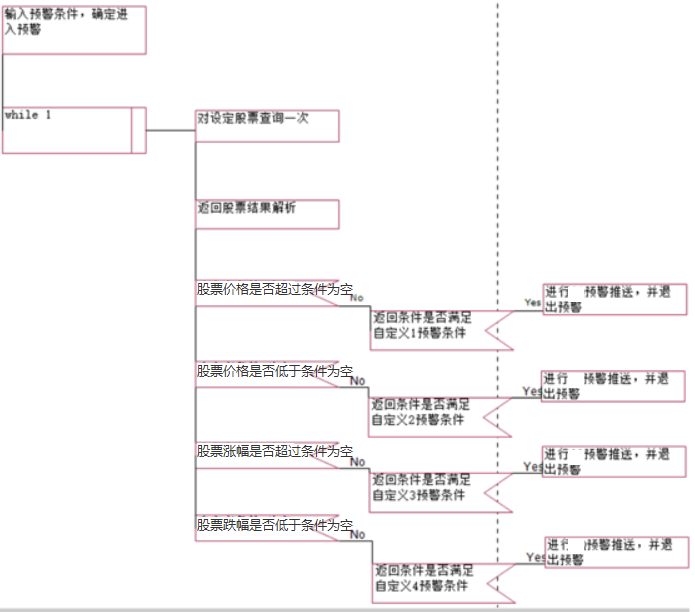
Else if（传入数据为小数）

讲小数改为百分数形式，并以绿色字体输出

Else讲数字用绿色字体输出

End if

### 微信预警推送



PAD图

Procedure 预警推送 is

While(1)

对该股票查询一次

对返回结果解析

If(股票价格是否超过条件为空)

Null

else if(返回结果中的值满足股票价格是否超过条件)

进行微信推送预警

break

else

null

If(股票价格是否低于条件为空)

Null

else if(返回结果中的值满足股票价格是否低于条件)

进行微信推送预警

break

else

null

If(股票价格涨幅是否超过为空)

Null

else if(返回结果中的值满足股票价格涨幅是否超过)

进行微信推送预警

break

else

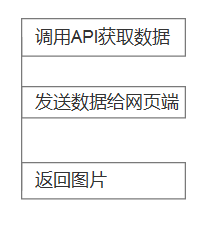
null

以此类推

每隔一定时间咨询一次

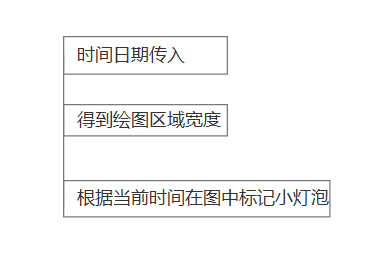
End 预警推送

### 个股实时走势图



PAD 图

### 预警灯泡



PAD 图

## 存储分配

本程序进行编码所用编程语言为C#，直接内存分配由C#运行时分配。

本组件内所以来的变量、结构要求全部在组件元素内申明。

## 限制条件

当用户输入的是不符合规则的代码时，则无法进行咨询。

## 测试计划

* 技术要点：成功显示股票咨询结果，进行微信预警推送。
* 输入数据：阿里云返回的JSON格式的匹配结果。
* 预期结果：将返回的结果解析然后输出在各项里面。
* 进度安排：一周之内完成。
* 人员职责：完成此模块的编码、测试、调试。
* 设备条件：装有Visual studio的电脑。

# 详细的实现计划

## 模块分类

本项目分为三个模块：

1.股票咨询模块

2.预警信息自定义模块

3.微信推送预警模块

## 代码设计分工

股票咨询模块：骆佳俊

预警信息自定义模块 ：吕迪

微信推送预警模块：徐双铅

## 代码实现分工

股票咨询模块：吕迪

预警信息自定义模块 ：徐双铅

微信推送预警模块：骆佳俊

## 代码测试分工

股票咨询模块：徐双铅

预警信息自定义模块 ：骆佳俊

微信推送预警模块：吕迪

# 详细的实现计划

## 界面设计

<UserControl

x:Class="SimpleTracer.Controls.AppSettings"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:com="clr-namespace:SimpleTracer.Commands"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:SimpleTracer"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:md="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:v="clr-namespace:SimpleTracer.Validation"

x:Name="control"

d:DesignHeight="600"

d:DesignWidth="800"

FontSize="16"

Loaded="control\_Loaded"

mc:Ignorable="d">

<UserControl.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<ResourceDictionary Source="/SimpleTracer;component/Resources/Styles.xaml" />

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</UserControl.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<StackPanel>

<md:Card Margin="5">

<GroupBox>

<GroupBox.Header>

<StackPanel Orientation="Horizontal" TextElement.Foreground="White">

<md:PackIcon

Width="24"

Height="24"

Margin="0,0,8,0"

VerticalAlignment="Center"

Kind="VolumeHigh" />

<TextBlock Style="{DynamicResource MaterialDesignTitleTextBlock}">声音设置</TextBlock>

</StackPanel>

</GroupBox.Header>

<Grid Margin="30,10">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="Auto" />

</Grid.ColumnDefinitions>

<TextBlock>预警触发时启用提示音</TextBlock>

<ToggleButton Grid.Column="1" IsChecked="{Binding EnableSound, ElementName=control, Mode=TwoWay}" />

</Grid>

</GroupBox>

</md:Card>

<md:Card Margin="5">

<GroupBox>

<GroupBox.Header>

<StackPanel Orientation="Horizontal" TextElement.Foreground="White">

<md:PackIcon

Width="24"

Height="24"

Margin="0,0,8,0"

VerticalAlignment="Center"

Kind="Publish" />

<TextBlock Style="{DynamicResource MaterialDesignTitleTextBlock}">推送设置</TextBlock>

</StackPanel>

</GroupBox.Header>

<Grid Margin="20,10">

<StackPanel>

<Grid Margin="10">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="Auto" />

</Grid.ColumnDefinitions>

<TextBlock>预警触发时启用消息推送</TextBlock>

<ToggleButton Grid.Column="1" IsChecked="{Binding EnablePush, ElementName=control, Mode=TwoWay}" />

</Grid>

<TextBox

Margin="10"

md:HintAssist.Hint="Server酱 API Key"

Style="{StaticResource MaterialDesignFloatingHintTextBox}">

<TextBox.Text>

<Binding

ElementName="control"

NotifyOnValidationError="True"

Path="PushKey">

<Binding.ValidationRules>

<v:RegexValidationRule RegexPattern="^[A-Za-z0-9]\*$" />

</Binding.ValidationRules>

</Binding>

</TextBox.Text>

</TextBox>

<Separator Style="{StaticResource MaterialDesignDarkSeparator}" />

<Grid HorizontalAlignment="Center">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<Image

Height="160"

Cursor="Hand"

MouseLeftButtonDown="Image\_MouseLeftButtonDown"

Source="/SimpleTracer;component/Resources/sc\_qrcode.png"

ToolTip="http://sc.ftqq.com/" />

<FlowDocumentScrollViewer

Grid.Column="1"

MaxWidth="600"

VerticalAlignment="Center"

VerticalScrollBarVisibility="Disabled">

<FlowDocumentScrollViewer.Document>

<FlowDocument FontFamily="Microsoft YaHei UI" PagePadding="0">

<List MarkerStyle="Decimal">

<ListItem>

<Paragraph>点击或者扫描二维码，打开绑定页面</Paragraph>

</ListItem>

<ListItem>

<Paragraph>使用Github账号登录并授权Server酱</Paragraph>

</ListItem>

<ListItem>

<Paragraph>填写API Key</Paragraph>

</ListItem>

<ListItem>

<Paragraph>绑定微信公众号</Paragraph>

</ListItem>

</List>

</FlowDocument>

</FlowDocumentScrollViewer.Document>

</FlowDocumentScrollViewer>

</Grid>

</StackPanel>

</Grid>

</GroupBox>

</md:Card>

</StackPanel>

<Separator

Margin="8,0"

VerticalAlignment="Bottom"

Style="{StaticResource MaterialDesignDarkSeparator}" />

<Grid Grid.Row="1">

<StackPanel HorizontalAlignment="Right" Orientation="Horizontal">

<Button

x:Name="clear"

Click="clear\_Click"

Content="清空"

Style="{DynamicResource dialogButtonStyle}" />

<Button

x:Name="save"

Click="save\_Click"

Content="保存"

Style="{DynamicResource dialogDefaultButtonStyle}" />

</StackPanel>

</Grid>

</Grid>

</UserControl>

<UserControl

x:Class="SimpleTracer.Controls.GlobalView"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:SimpleTracer.Controls"

xmlns:m="clr-namespace:SimpleTracer.Models"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:md="http://materialdesigninxaml.net/winfx/xaml/themes"

d:DesignHeight="300"

d:DesignWidth="300"

Loaded="UserControl\_Loaded"

mc:Ignorable="d">

<UserControl.Resources>

<DataTemplate DataType="{x:Type m:StockInfo}">

<StackPanel Orientation="Horizontal">

<md:PackIcon Kind="ChevronRight" />

<TextBlock Text="{Binding Name}" />

</StackPanel>

</DataTemplate>

</UserControl.Resources>

<Grid>

<md:Card

Margin="10"

HorizontalAlignment="Center"

VerticalAlignment="Stretch">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="\*" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="200" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<TextBlock

Margin="10"

HorizontalAlignment="Center"

VerticalAlignment="Center"

FontSize="36"

Foreground="{DynamicResource SecondaryAccentBrush}">

大盘走势

</TextBlock>

<ListBox

x:Name="list"

Grid.Row="1"

Background="{DynamicResource PrimaryHueLightBrush}"

Foreground="{DynamicResource PrimaryHueLightForegroundBrush}"

ItemsSource="{Binding Indexes, RelativeSource={RelativeSource FindAncestor, AncestorType={x:Type local:GlobalView}}}"

SelectedIndex="0"

SelectedItem="{Binding CurrentItem, Mode=OneWayToSource, RelativeSource={RelativeSource FindAncestor, AncestorType={x:Type local:GlobalView}}}" />

<WebBrowser

x:Name="browser"

Grid.RowSpan="2"

Grid.Column="1"

Tag="{Binding SelectedItem, ElementName=list}" />

</Grid>

</md:Card>

</Grid>

</UserControl>

<UserControl

x:Class="SimpleTracer.Controls.ParameterAlert"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:com="clr-namespace:SimpleTracer.Commands"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:SimpleTracer.Controls"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:md="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:mdl="clr-namespace:SimpleTracer.Models"

xmlns:validation="clr-namespace:SimpleTracer.Validation"

x:Name="userControl"

d:DesignHeight="600"

d:DesignWidth="800"

mc:Ignorable="d">

<UserControl.CommandBindings>

<CommandBinding

x:Name="saveCommand"

CanExecute="Command\_CanExecute"

Command="{x:Static ApplicationCommands.Save}"

Executed="saveCommand\_Executed" />

<CommandBinding

x:Name="clearCommand"

CanExecute="Command\_CanExecute"

Command="{x:Static com:CollectionCommands.Clear}"

Executed="clearCommand\_Executed" />

</UserControl.CommandBindings>

<UserControl.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<ResourceDictionary Source="/SimpleTracer;component/Resources/Styles.xaml" />

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</UserControl.Resources>

<Grid>

<Grid>

<GroupBox>

<GroupBox.Header>

<Grid>

<StackPanel Orientation="Horizontal">

<Button Style="{StaticResource backButtonStyle}" />

<TextBlock

VerticalAlignment="Center"

Foreground="White"

Style="{DynamicResource MaterialDesignTitleTextBlock}">

预警规则

</TextBlock>

</StackPanel>

</Grid>

</GroupBox.Header>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<Grid

x:Name="container"

Margin="80,20"

DataContext="{Binding CurrentAlertConditions, ElementName=userControl}"

Grid.IsSharedSizeScope="True">

<Grid.RowDefinitions>

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="Auto" />

</Grid.ColumnDefinitions>

<Grid.Resources>

<Style TargetType="StackPanel">

<Setter Property="Orientation" Value="Horizontal" />

</Style>

<Style BasedOn="{StaticResource {x:Type TextBox}}" TargetType="TextBox">

<Setter Property="VerticalAlignment" Value="Center" />

<Setter Property="MinWidth" Value="200" />

<Setter Property="MaxWidth" Value="500" />

<Setter Property="Margin" Value="5,0" />

</Style>

<Style TargetType="TextBlock">

<Setter Property="VerticalAlignment" Value="Center" />

<Setter Property="Margin" Value="10,0,5,0" />

</Style>

<Style BasedOn="{StaticResource MaterialDesignSwitchToggleButton}" TargetType="CheckBox">

<Setter Property="Grid.Column" Value="1" />

<Setter Property="Margin" Value="20,0" />

<Setter Property="ToolTip" Value="启用/禁用参数" />

</Style>

</Grid.Resources>

<StackPanel>

<TextBlock Text="股价超过 " />

<!-- 下面这些神奇的绑定表达式是简写形式，理解请注意前面元素的DataContext -->

<TextBox x:Name="priceOver" Foreground="Red">

<TextBox.Text>

<Binding NotifyOnValidationError="True" Path="[0].alertValue">

<Binding.ValidationRules>

<validation:NoNegativeValidationRule />

</Binding.ValidationRules>

</Binding>

</TextBox.Text>

</TextBox>

<TextBlock>元</TextBlock>

</StackPanel>

<CheckBox IsChecked="{Binding [0].enabled}" />

<StackPanel Grid.Row="1">

<TextBlock Text="股价低于 " />

<TextBox x:Name="priceUnder" Foreground="LimeGreen">

<TextBox.Text>

<Binding NotifyOnValidationError="True" Path="[1].alertValue">

<Binding.ValidationRules>

<validation:NoNegativeValidationRule />

</Binding.ValidationRules>

</Binding>

</TextBox.Text>

</TextBox>

<TextBlock>元</TextBlock>

</StackPanel>

<CheckBox Grid.Row="1" IsChecked="{Binding [1].enabled}" />

<StackPanel Grid.Row="2">

<TextBlock Text="日涨幅超过 " />

<TextBox x:Name="dailyRateUp" Foreground="Red">

<TextBox.Text>

<Binding NotifyOnValidationError="True" Path="[2].alertValue">

<Binding.ValidationRules>

<validation:NoNegativeValidationRule />

</Binding.ValidationRules>

</Binding>

</TextBox.Text>

</TextBox>

<TextBlock>%</TextBlock>

</StackPanel>

<CheckBox Grid.Row="2" IsChecked="{Binding [2].enabled}" />

<StackPanel Grid.Row="3">

<TextBlock Text="日跌幅超过 " />

<TextBox x:Name="dailyRateDown" Foreground="LimeGreen">

<TextBox.Text>

<Binding NotifyOnValidationError="True" Path="[3].alertValue">

<Binding.ValidationRules>

<validation:NoNegativeValidationRule />

</Binding.ValidationRules>

</Binding>

</TextBox.Text>

</TextBox>

<TextBlock>%</TextBlock>

</StackPanel>

<CheckBox Grid.Row="3" IsChecked="{Binding [3].enabled}" />

<StackPanel Grid.Row="4">

<TextBlock>5分钟涨幅超过</TextBlock>

<TextBox x:Name="fiveMinRateUp" Foreground="Red">

<TextBox.Text>

<Binding NotifyOnValidationError="True" Path="[4].alertValue">

<Binding.ValidationRules>

<validation:NoNegativeValidationRule />

</Binding.ValidationRules>

</Binding>

</TextBox.Text>

</TextBox>

<TextBlock>%</TextBlock>

</StackPanel>

<CheckBox Grid.Row="4" IsChecked="{Binding [4].enabled}" />

<StackPanel Grid.Row="5">

<TextBlock>5分钟跌幅超过</TextBlock>

<TextBox x:Name="fiveMinRateDown" Foreground="LimeGreen">

<TextBox.Text>

<Binding NotifyOnValidationError="True" Path="[5].alertValue">

<Binding.ValidationRules>

<validation:NoNegativeValidationRule />

</Binding.ValidationRules>

</Binding>

</TextBox.Text>

</TextBox>

<TextBlock>%</TextBlock>

</StackPanel>

<CheckBox Grid.Row="5" IsChecked="{Binding [5].enabled}" />

</Grid>

<Separator

Margin="8,0"

VerticalAlignment="Bottom"

Style="{StaticResource MaterialDesignDarkSeparator}" />

<Grid Grid.Row="1">

<StackPanel HorizontalAlignment="Right" Orientation="Horizontal">

<Button

x:Name="clear"

Command="{x:Static com:CollectionCommands.Clear}"

Content="清空"

Style="{DynamicResource dialogButtonStyle}" />

<Button

x:Name="save"

Command="{x:Static ApplicationCommands.Save}"

Content="保存"

Style="{DynamicResource dialogDefaultButtonStyle}" />

</StackPanel>

</Grid>

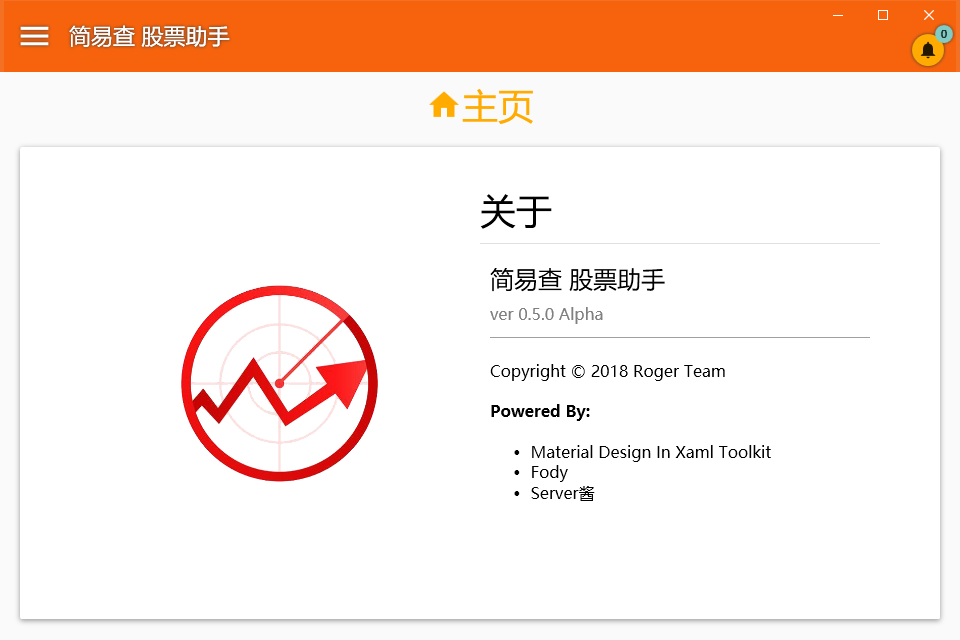
</Grid>

</GroupBox>

</Grid>

</Grid>

</UserControl>





## 关键算法设计

/// <summary>

/// 股价超过指定值的报警规则

/// </summary>

public class PriceOverChecker : ParamCheckerBase

{

public override bool Check(Dictionary<double, RealTimeQuotationModel> data, AlertCondition condition, out double alertValue)

{

alertValue = 0;

if (data.Count > 0)

{

var last = data[data.Keys.Max()];

alertValue = last.last\_px;

return last.last\_px >= condition.alertValue;

}

return false;

}

}

namespace SimpleTracer.Validation

{

/// <summary>

/// 通过判断数字是否为负验证值的规则

/// </summary>

public class NoNegativeValidationRule : ValidationRule

{

public override ValidationResult Validate(object value, CultureInfo cultureInfo)

{

if (value is string s)

{

if (double.TryParse(s, out var v))

{

return v >= 0 ? new ValidationResult(true, null)

: new ValidationResult(false, "值不能是负数");

}

}

return new ValidationResult(false, "值不是有效的数字");

}

}

}

Text="{Binding Tag, RelativeSource={RelativeSource FindAncestor, AncestorType={x:Type ContentControl}}}" />

<Run Text="{Binding px\_change, Converter={StaticResource covChange}}" />

<Run Text="{Binding px\_change\_rate, Converter={StaticResource covChange}, StringFormat=({0}), ConverterParameter=%}" />

各种绑定。