M05 - Working with APIs (UWP)

Author: Yi Chen
Date: 2021.02.26

M05 - Working with APIs (UWP)

Table of Contents

List of Tables

List of Figures

1. Working with APIs (UWP)

- Using Universal Windows Platform (UWP) create an app which utilizes data from a Tushare API.

2. Project Requirements

- Use a free public API to get data for your application.
  - It is up to you to decide which API you want to use.
  - If your API requires a key, make sure the key is in your code.

Derived Requirements

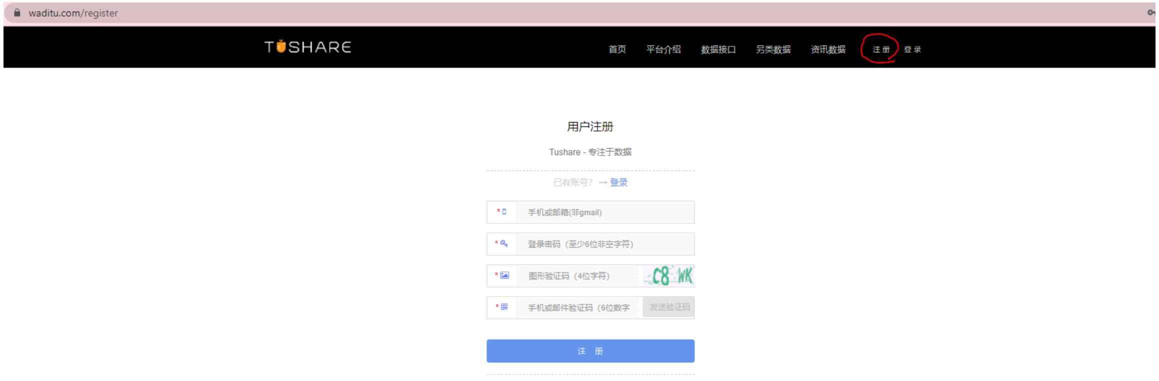
- ✓ A prepared PDF document with your commented code.
- ✓ Document should have relevant meta information and be well formatted.

3. Tushare API Introduction

- Site: https://waditu.com/document/1
- ❖ Tushare API is including global stocks, funds, futures, bonds, foreign exchange, and Guidepoin big data.
- ❖ Additionally, the digital currency market and other blockchain data of the full data category of a financial big data platform.
- ❖ It is for all kinds of financial investments and researchers to provide applicable data and tools.

3.1. Register:

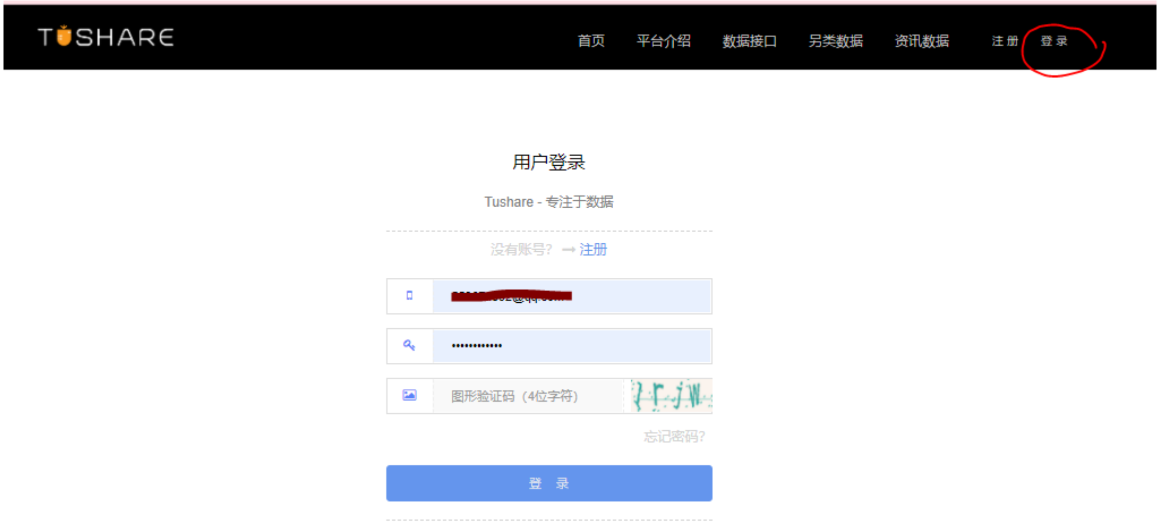
- ➢ The site using the system to control the access of data, if you want to access data, to complete registration to the URL, https://tushare.pro/register.



- ➢ After the registration is completed. You may need to copy the Token from your personal homepage. This Token will be used in future visits.

- The steps:

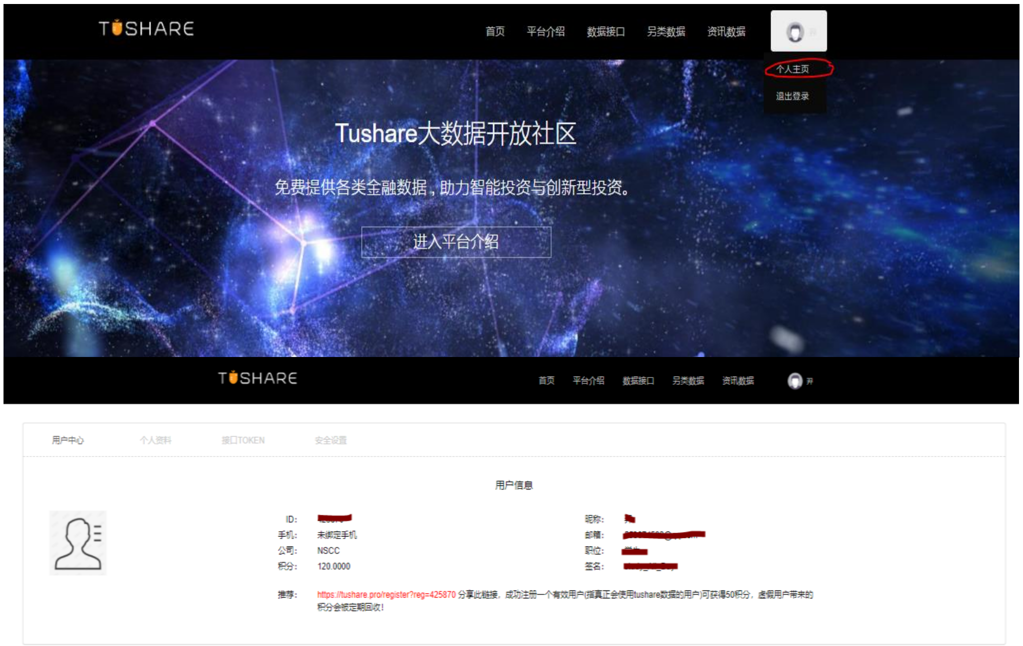
- ✓ logged in, click on the upper right corner -> personal home page



[redacted]

••••••••••••

- ✓ Click "Interface Token" in "User Center"



Tushare大数据开放社区

免费提供各类金融数据，助力智能投资与创新型投资。

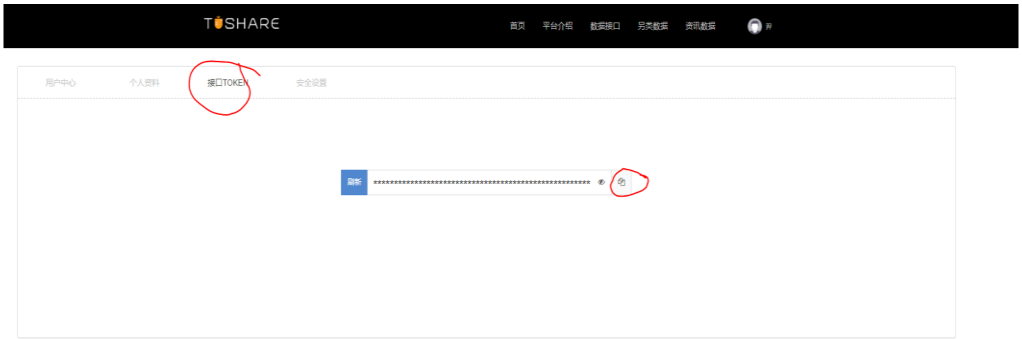
用户信息

ID: [redacted]
手机: 未绑定手机
公司: NSCC
积分: 120.0000

昵称: [redacted]
邮箱: [redacted]
职位: [redacted]
签名: [redacted]

推荐: https://tushare.pro/register?reg=425870 分享此链接，成功注册一个有效用户(指真正会使用tushare数据的用户)可获得50积分，虚假用户带来的积分会被定期回收!

✓ You can copy the token by clicking the copy button on the right.



****

3.2. Data Acquisition Mode

- There are four ways to get data from the Tushare API
  - ➢ Fetching data over HTTP Restful API
  - Fetch data through the Python SDK
  - Data is obtained through the MATLAB SDK
  - Fetch data through R language

I used the first way which is Fetching data over HTTP, and in this document, I will introduce you the method.

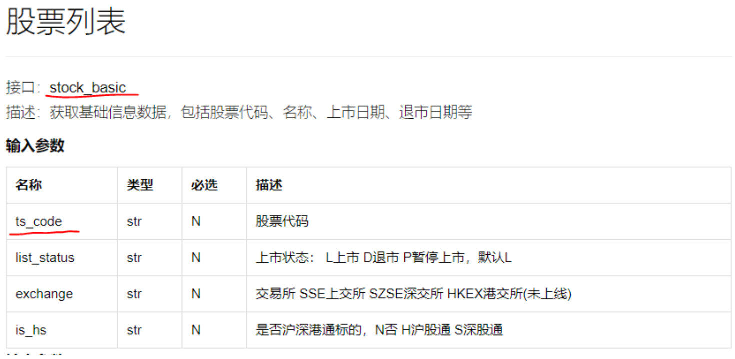
3.2.1. Fetching data over HTTP Restful API

- HTTP Restful API Introduction:

- Tushare uses the mechanism of POST to obtain the HTTP data. You can get the data you want by submitting the JSON body parameter.

  - Input parameter

    - api_name: Interface name, such as “stock_basic”.
    - token: User-unique identifier that can obtain by logging into the pro website.
    - params: Interface parameters, such as “start_date” and “end_date” in the daily interface
    - fields: A list of fields used by the interface to retrieve the specified fields, separated by commas, such as “open, high, low, close”.



股票列表

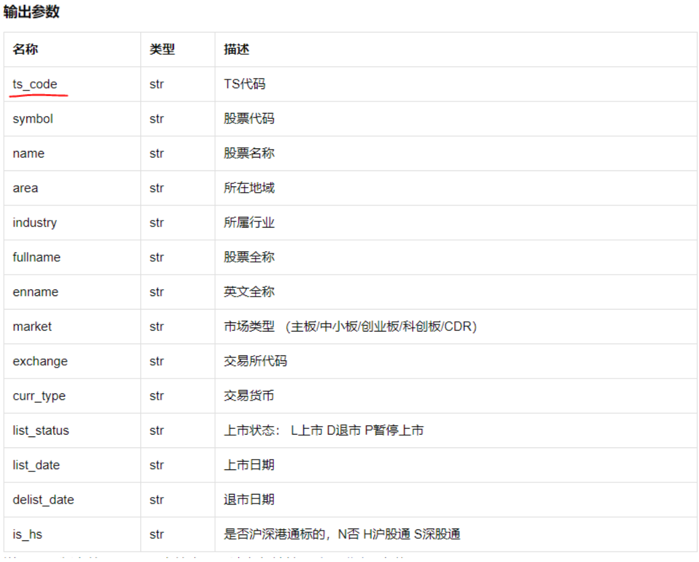
接口：stock_basic
描述：获取基础信息数据，包括股票代码、名称、上市日期、退市日期等

输入参数

名称	类型	必选	描述
ts_code	str	N	股票代码
list_status	str	N	上市状态： L上市 D退市 P暂停上市，默认L
exchange	str	N	交易所 SSE上交所 SZSE深交所 HKEX港交所(未上线)
is_hs	str	N	是否沪深港通标的，N否 H沪股通 S深股通

  - output parameter

    - code: Interface return code, 2002 indicates permission issue.
    - msg: error message, such as "system internal error", "no permissions", etc.
    - data: Data contains Fields and Items fields, which are fields and data contents, respectively.



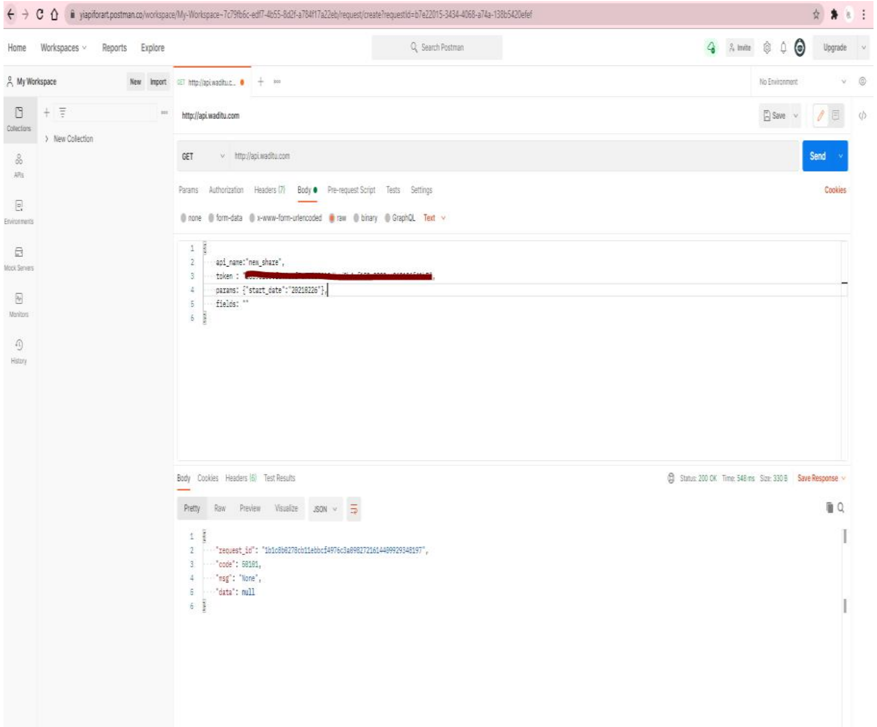
输出参数

名称	类型	描述
ts_code	str	TS代码
symbol	str	股票代码
name	str	股票名称
area	str	所在地域
industry	str	所属行业
fullname	str	股票全称
enname	str	英文全称
market	str	市场类型 （主板/中小板/创业板/科创板/CDR）
exchange	str	交易所代码
curr_type	str	交易货币
list_status	str	上市状态： L上市 D退市 P暂停上市
list_date	str	上市日期
delist_date	str	退市日期
is_hs	str	是否沪深港通标的，N否 H沪股通 S深股通

3.2.2. API tools for quick detection

- If you want to get a quick and easy look at the effects of the data API and check availability, but don't want to write code, PostMan might be a useful tool.

- Run PostMan, select POST, enter http://api.waditu.com in the API address bar, then click Body below, and enter the JSON format parameter.



GET http://api.waditu.com

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL Text

```
{
  api_name:"new_share",
  token : [redacted],
  params: {"start_date":"20210226"},
  fields: ""
}
```

Body Cookies Headers (6) Test Results — Status: 200 OK Time: 548 ms Size: 330 B

JSON

```
{
  "request_id": "1b1c8b0278cb11ebbcf4976c3a8982721614409929348197",
  "code": 50101,
  "msg": "None",
  "data": null
}
```

4. Implementation

- *Install packages in Visual Studio Manage NeGet Packages for solution:*
  - *Microsoft.Extensions.Http*
  - *Newtonsoft.JsonImplement*
  - *Microsoft.Toolkit.Uwp.UI.Controls.DataGrid*
  - *Microsoft.NETCore.UniversalWindowsPlatform*

4.1. Implement calls to Rest Web services

- *Interface:*
  - *IHttpClientUtility*
- *Class:*
  - *HttpClientUtility*
  - *JsonContent*

Figure 1: Encapsulate methods to implement calls to REST Web services

Interface

```
    public interface IHttpClientUtility
    {
          string HttpClientPost(string url, object datajson);
    }
```

HttpClientUtility

```
using System;
using System.Threading.Tasks;
using System.Net.Http;
using System.Net.Http.Headers;

namespace StockAPI
{
    public class HttpClientUtility : IHttpClientUtility
    {


        public HttpClientUtility()
        {

        }
        public string HttpClientPost(string url, object datajson)
        {
            using (HttpClient httpClient = new HttpClient()) //http Object
            {
                httpClient.DefaultRequestHeaders.Accept.Clear();
                httpClient.DefaultRequestHeaders.Accept.Add(new 
MediaTypeWithQualityHeaderValue("application/json"));
                httpClient.Timeout = new TimeSpan(0, 0, 5);
                //Convert to the format required for the link
                HttpContent httpContent = new JsonContent(datajson);

                //request
                HttpResponseMessage response = httpClient.PostAsync(url, 
httpContent).Result;
                if (response.IsSuccessStatusCode)
                {
                    Task<string> t = response.Content.ReadAsStringAsync();
                    return t.Result;
                }
                throw new Exception("failed to call");
            }

        }
    }
}
```

JsonContent

```
using System.Text;
using Newtonsoft.Json;
using System.Net.Http;

namespace StockAPI
{
    public class JsonContent : StringContent
    {
        /*get SerializeObject value*/
        public JsonContent(object value)
            : base(JsonConvert.SerializeObject(value), Encoding.UTF8,
                "application/json")
        {
        }

        public JsonContent(object value, string mediaType)
            : base(JsonConvert.SerializeObject(value), Encoding.UTF8, mediaType)
        {
        }
    }
}
```

4.2. Encapsulates calls to the Tushare API

- *Class:*
  - *TuShareUtility*

Table 1: Encapsulates calls to the Tushare API

TuShareUtility

```
using System;
using System.Collections.Generic;
using Newtonsoft.Json;
using System.Data;
```

```
namespace StockAPI
{
    public class TuShareUtility
    {
        private IHttpClientUtility _httpClientUtility;
        private string _url = "http://api.waditu.com/";

        public TuShareUtility(IHttpClientUtility httpClientUtility)
        {
            _httpClientUtility = httpClientUtility;
        }

        /// <summary>
        /// Call TuShare API
        /// </summary>
        /// <param name="apiName"></param>
        /// <param name="parmaMap"></param>
        /// <param name="fields"></param>
        /// <returns></returns>
        public DataTable GetData(string apiName, Dictionary<string, string>
parmaMap, params string[] fields)
        {
            var tuShareParamObj = new TuShareParamObj() { ApiName = apiName,
Params = parmaMap, Fields = string.Join(",", fields) };

            //Http call HttpClientUtility class with in HttpClientPost
methods
            var result = _httpClientUtility.HttpClientPost(_url,
tuShareParamObj);

            //Serializes the returned result into an object
            var desResult =
JsonConvert.DeserializeObject<TuShareResult>(result);

            //If the call fails, an exception is thrown
            if (!string.IsNullOrEmpty(desResult.Msg))
                throw new Exception(desResult.Msg);

            //The return result is divided into two parts, one is the column
header information
            //and the other is the data itself, from which the DataTable can
be built
            DataTable dt = new DataTable();
            foreach (var dataField in desResult.Data.Fields)
            {
                dt.Columns.Add(dataField);
            }

            foreach (var dataItemRow in desResult.Data.Items)
            {
                var newdr = dt.NewRow();
                for (int i = 0; i < dataItemRow.Length; i++)
                {
                    newdr[i] = dataItemRow[i];
                }

                dt.Rows.Add(newdr);
            }
            return dt;
        }

        private class TuShareParamObj
        {
            [JsonProperty("api_name")]
            public string ApiName { get; set; }

            [JsonProperty("token")]
            public string Token { get; } = 
"a0b9d10ecfa79acf7a93d28110dbcd0b4cf152c9892ee363106f41b7";//my Token

            [JsonProperty("params")]
            public Dictionary<string, string> Params { get; set; }

            [JsonProperty("fields")]
            public string Fields { get; set; }
        }

        private class TuShareData
        {
            [JsonProperty("fields")]
            public string[] Fields { get; set; }

            [JsonProperty("items")]
            public string[][] Items { get; set; }
        }

        private class TuShareResult
        {
            [JsonProperty("code")]
            public string Code { get; set; }

            [JsonProperty("msg")]
            public string Msg { get; set; }

            [JsonProperty("data")]
            public TuShareData Data { get; set; }
        }
    }
}
```

4.3. Show Date in UWP page

- *MainPage :*
    - *DataGrid*
    - *Click_Btn_Click*

    - *Methods:*
        - *FillDataGrid -> Fill DataTable to DataGrid*

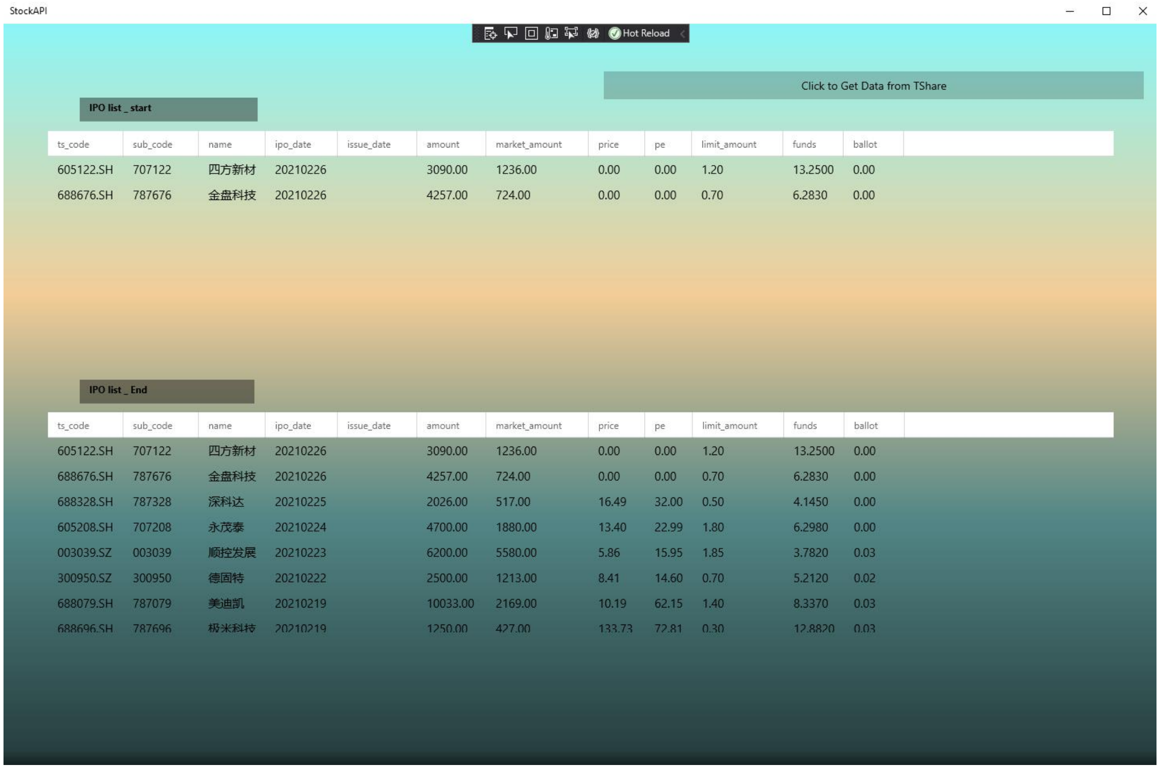
Table 2: Visulaze data in UWP page

MainPage

```
using System.Collections.Generic;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Data;
using Microsoft.Toolkit.Uwp.UI.Controls;
using System.Collections.ObjectModel;
using System.Data;




// The Blank Page item template is documented at
https://go.microsoft.com/fwlink/?LinkId=402352&clcid=0x409

namespace StockAPI
{
    /// <summary>
    /// An empty page that can be used on its own or navigated to within a Frame.
    /// </summary>
    ///  Author : Yi Chen
    public sealed partial class MainPage : Page
    {

        public MainPage()
        {
            this.InitializeComponent();
        }

        private void Click_Btn_Click(object sender, RoutedEventArgs e)
        {
            var tuShareUtility = new TuShareUtility(new HttpClientUtility());
            Dictionary<string, string> p = new Dictionary<string, string>();

            p["end_date"] = "20210226";
            var table = tuShareUtility.GetData("new_share", p, "");

            p["start_date"] = "20210226";
            var table2 = tuShareUtility.GetData("new_share", p, "");

            dataGrid.ItemsSource = table.DefaultView;
            dataGrid1.ItemsSource = table2.DefaultView;

            FillDataGrid(table, dataGrid);
            FillDataGrid(table2, dataGrid1);
        }

        public static void FillDataGrid(DataTable table, DataGrid grid)
        {
            grid.Columns.Clear();
            grid.AutoGenerateColumns = false;
            for (int i = 0; i < table.Columns.Count; i++)
            {
                grid.Columns.Add(new DataGridTextColumn()
                {
                    Header = table.Columns[i].ColumnName,
                    Binding = new Binding { Path = new PropertyPath("[" +
i.ToString() + "]") }
                });
            }

            var collection = new ObservableCollection<object>();
            foreach (DataRow row in table.Rows)
            {
                collection.Add(row.ItemArray);
            }

            grid.ItemsSource = collection;
        }


    }
}
```

4.3.1. The Data Table Result

- Click Button:
    - Click to Get Data from Tushare
        - IPO List Data:
            - ✓ start_date : 2021-02-26
            - ✓ end_date: 2021-02-26

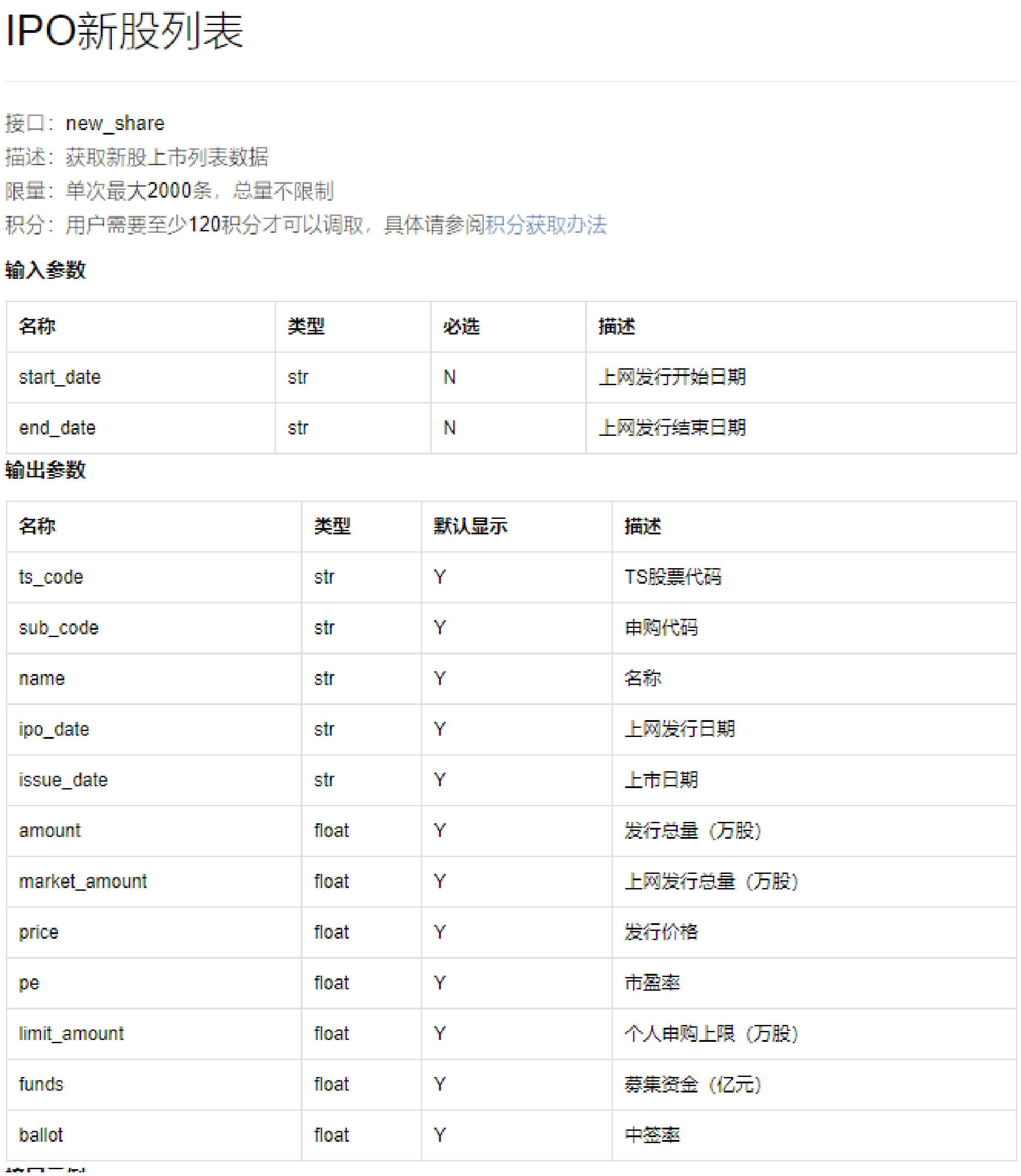


IPO list _ start

ts_code	sub_code	name	ipo_date	issue_date	amount	market_amount	price	pe	limit_amount	funds	ballot
605122.SH	707122	四方新材	20210226		3090.00	1236.00	0.00	0.00	1.20	13.2500	0.00
688676.SH	787676	金盘科技	20210226		4257.00	724.00	0.00	0.00	0.70	6.2830	0.00

IPO list _ End

ts_code	sub_code	name	ipo_date	issue_date	amount	market_amount	price	pe	limit_amount	funds	ballot
605122.SH	707122	四方新材	20210226		3090.00	1236.00	0.00	0.00	1.20	13.2500	0.00
688676.SH	787676	金盘科技	20210226		4257.00	724.00	0.00	0.00	0.70	6.2830	0.00
688328.SH	787328	深科达	20210225		2026.00	517.00	16.49	32.00	0.50	4.1450	0.00
605208.SH	707208	永茂泰	20210224		4700.00	1880.00	13.40	22.99	1.80	6.2980	0.00
003039.SZ	003039	顺控发展	20210223		6200.00	5580.00	5.86	15.95	1.85	3.7820	0.03
300950.SZ	300950	德固特	20210222		2500.00	1213.00	8.41	14.60	0.70	5.2120	0.02
688079.SH	787079	美迪凯	20210219		10033.00	2169.00	10.19	62.15	1.40	8.3370	0.03
688696.SH	787696	极米科技	20210219		1250.00	427.00	133.73	72.81	0.30	12.8820	0.03



IPO新股列表

接口：new_share
描述：获取新股上市列表数据
限量：单次最大2000条，总量不限制
积分：用户需要至少120积分才可以调取，具体请参阅积分获取办法

输入参数

名称	类型	必选	描述
start_date	str	N	上网发行开始日期
end_date	str	N	上网发行结束日期

输出参数

名称	类型	默认显示	描述
ts_code	str	Y	TS股票代码
sub_code	str	Y	申购代码
name	str	Y	名称
ipo_date	str	Y	上网发行日期
issue_date	str	Y	上市日期
amount	float	Y	发行总量（万股）
market_amount	float	Y	上网发行总量（万股）
price	float	Y	发行价格
pe	float	Y	市盈率
limit_amount	float	Y	个人申购上限（万股）
funds	float	Y	募集资金（亿元）
ballot	float	Y	中签率

Table 4: MainPage.xmal

MainPage.Xaml

```
<Page
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="using:StockAPI"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:Custom="using:Microsoft.Toolkit.Uwp.UI.Controls"
    xmlns:controls="using:Microsoft.Toolkit.Uwp.UI.Controls"

    x:Class="StockAPI.MainPage"
    mc:Ignorable="d"
    Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">

    <Grid>
        <Grid.Background>
            <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
                <GradientStop Color="Black" Offset="1"/>
                <GradientStop Color="#FFF3CD97" Offset="0.358"/>
                <GradientStop Color="#FF538787" Offset="0.653"/>
                <GradientStop Color="#FF263F3F" Offset="0.961"/>
                <GradientStop Color="#FF8DF5F5" Offset="0.005"/>
            </LinearGradientBrush>
        </Grid.Background>
        <Grid.FocusVisualSecondaryBrush>
            <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
                <GradientStop Color="Black"/>
                <GradientStop Color="#FFCDC35E" Offset="1"/>
            </LinearGradientBrush>
        </Grid.FocusVisualSecondaryBrush>
        <Button x:Name="Click_Btn" Content="Click to Get Data from TShare"
Margin="760,60,0,0" VerticalAlignment="Top" Height="35" Width="680"
Click="Click_Btn_Click"/>

        <controls:DataGrid x:Name="dataGrid"
                       Margin="60,489,60,186"
                       AutoGenerateColumns="True">
        </controls:DataGrid>

        <controls:DataGrid x:Name="dataGrid1"
                       Margin="60,135,60,560"
                       AutoGenerateColumns="True"/>
        <TextBox x:Name="Lab_01" HorizontalAlignment="Left" Margin="100,93,0,0"
Text="IPO list _ start" TextWrapping="Wrap" VerticalAlignment="Top" Height="30"
Width="224" FontSize="12" FontWeight="Bold" Foreground="Black"
FocusVisualPrimaryBrush="#FFFBF7F7" Background="#66020000"/>
        <TextBox x:Name="Lab_02" HorizontalAlignment="Left" Margin="100,448,0,0"
Text="IPO list _ End" TextWrapping="Wrap" VerticalAlignment="Top" Height="30"
Width="220" FontSize="12" FontWeight="Bold" Foreground="Black"
FocusVisualPrimaryBrush="#FFFBF7F7" Background="#66020000"/>




    </Grid>

</Page>
```