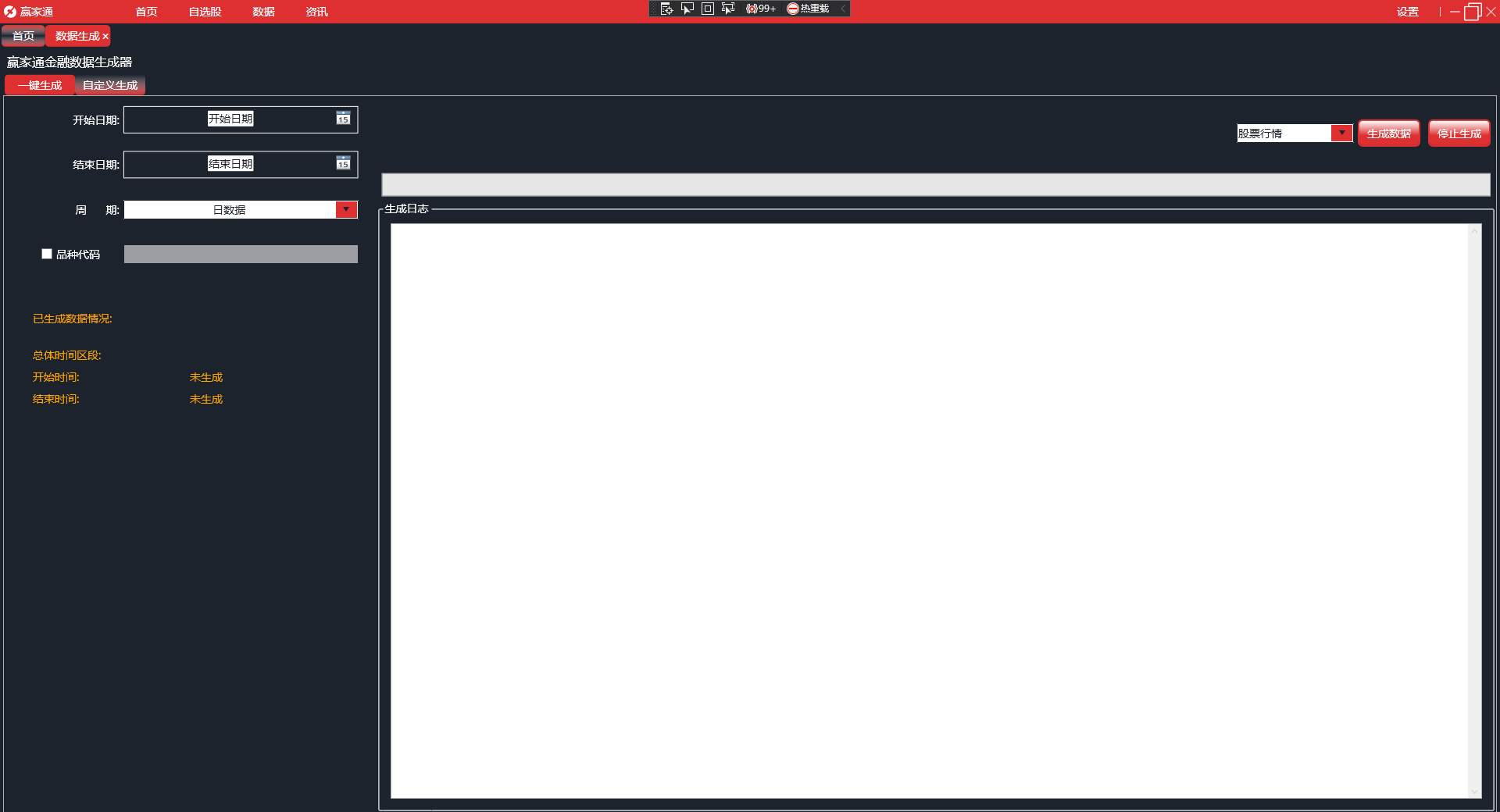
一、数据生成说明

* 打开WtaFinancePlatform\_WPF程序，点击**数据**->**数据生成**，进入**赢家通金融数据生成器**



* 在**赢家通金融数据生成器**主页，填写开始日期、结束日期、选择周期、生成数据类型，点击**生成数据**，即可生成数据。(生成数据类型为股票行情时会下载前复权、后复权、不复权的文件)。



* 开始日期：控制行情数据的开始日期；
* 结束日期：控制行情数据的结束日期；
* 周期：可选择日数据、五分钟数据；
* 生成数据类型：可选择股票行情、指数行情、基金行情、财务（利润表、现金流量表和资产负债表数据）；
* 股票行情：会依据周期生成前复权、后复权、不复权的股票数据
* 指数行情：会依据周期生成前复权、后复权、不复权指数行情
* 基金行情：基金数据还在爬取中。
* 财务：生成利润表、现金流量表和资产负债表数据。
* 左下角已生成数据情况，如果选择的文件类型全盘生成过就会显示总体时间区段时间。
* 点击“生成数据”，默认情况下是将所有股票的数据生成在.\Data\System\SZ或.\Data\System\SH目录下，含4000多支股票数据，生成数据的时间根据设置的条件来决定，时间较长，建议空闲时间批量生成；
* 如勾选了“股票代码”，并填写了要生成数据的股票代码，则可有针对性地对某一支股票生成数据。同时左下角会出现该个股的生成情况。

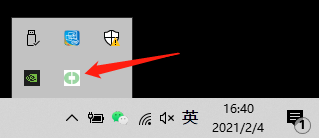
**文件生成路径**

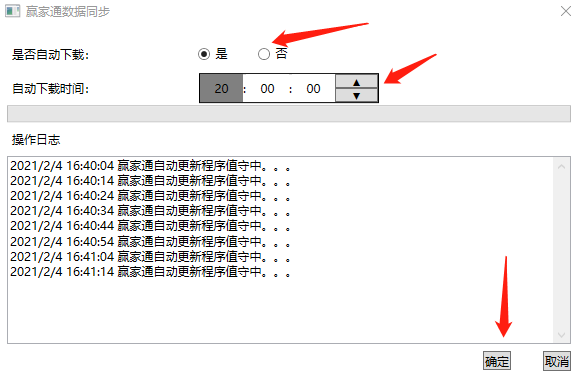
|  |  |  |  |
| --- | --- | --- | --- |
| 指数 | 日数据 | 不复权 | Data\System\INDEX\day\000001.dat |
| 指数 | 日数据 | 前复权 | Data\System\INDEX\day\000001.ldat |
| 指数 | 日数据 | 后复权 | Data\System\INDEX\day\000001.rdat |
| 指数 | 五分钟 | 不复权 | Data\System\INDEX\fzline\000001.fat |
| 指数 | 五分钟 | 前复权 | Data\System\INDEX\fzline\000001.lfat |
| 指数 | 五分钟 | 后复权 | Data\System\INDEX\fzline\000001.rfat |
| 股票 | 日数据 | 不复权 | Data\System\SH\day\000001.dat |
| 股票 | 日数据 | 前复权 | Data\System\SH\day\000001.ldat |
| 股票 | 日数据 | 后复权 | Data\System\SH\day\000001.rdat |
| 股票 | 五分钟 | 不复权 | Data\System\SZ\fzline\000001.fat |
| 股票 | 五分钟 | 前复权 | Data\System\SZ\fzline\000001.lfat |
| 股票 | 五分钟 | 后复权 | Data\System\SZ\fzline\000001.rfat |
| 财务 | / | / | Data\System\SH\finance\000001.fin |

1. 数据自动同步服务

数据自动同步服务为WtaSyncData项目，该项目在编辑器中启动无法正常同步数据，需要配合主程序打包安装后才能正常使用。

安装包位置为WtaFinancePlatform\setup\WtaFinancePlatformWin.msi，安装完成后桌面会创建，打开该快捷方式为主程序。自动同步服务在安装完成后会注入系统的注册表中。在每次开机时会自动启动。

该同步服务在系统托盘位置，，单机即可打开界面查看生产情况，可通过配置项来进行配置，数据同步服务会下载全部数据。



1. Python SDK使用说明

* 安装wtafinance库

pip install wtafinance

* 更新wtafinance

pip install --upgrade wtafinance

* 使用示例

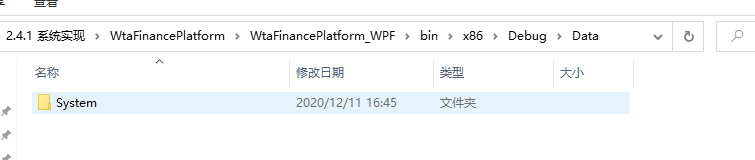
from wtafinance.stock import wta\_data

mainclass = wta\_data.wta\_file(**path**)

* **Path**路径说明

生成数据主程序的Data文件夹

例如：



**Path**为绝对路径，参考上图，path的值为：

D:\tby\_svn\wta\_finance\trunk\2 项目实施阶段\2.4 系统开发\2.4.1 系统实现\WtaFinancePlatform\WtaFinancePlatform\_WPF\bin\x86\Debug\Data

* 以下为mainclass里面获取数据的方法

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **方法名称** | **说明** | **参数** | **参数说明** | **是否必填** | **默认值** | | **类型** |
| get\_hist\_data | 获取股票行情数据 | code | 股票代码 | 是 |  | | str |
| start | 开始时间 | 是 |  | | str |
| end | 结束时间 | 否 | ‘’ | | str |
| fqt | 复权值（0不复权，1前复权，2后复权） | 否 | 0 | | str |
| ktype | 数据类型（5五分钟行情，D日行情） | 否 | day | | str |
| 示例：获取600000股票（2015-01-01至2021-01-01）日行情前复权数据 | | | | | |
| mainclass.get\_k\_data("600000","2015-01-01","2021-01-01",fqt="1",ktype="D") | | | | | |
| get\_index | 获取指数行情数据 | code | 指数代码 | 是 |  | str | |
| start | 开始时间 | 是 |  | str | |
| end | 结束时间 | 否 | ‘’ | str | |
| fqt | 复权值（0不复权，1前复权，2后复权） | 否 | 0 | str | |
| ktype | 数据类型（5五分钟行情，D日行情） | 否 | day | str | |
| 示例：获取000001指数（2015-01-01至2021-01-01）日行情前复权数据 | | | | | |
| mainclass.get\_Index\_data("600000","2015-01-01","2021-01-01",fqt="1",ktype="D") | | | | | |
| income | 获取非金融利润表财报数据 | code | 股票代码 | 是 |  | str | |
| start | 开始时间 | 是 |  | str | |
| end | 结束时间 | 否 | ‘’ | str | |
| 示例：获取600000股票（2015-01-01至2021-01-01）非金融利润表财报数据 | | | | | |
| mainclass.get\_IncomeStatement\_data("600000","2015-01-01","2021-01-01") | | | | | |
| cashflow | 获取非金融现金流量表财报数据 | code | 股票代码 | 是 |  | str | |
| start | 开始时间 | 是 |  | str | |
| end | 结束时间 | 否 | ‘’ | str | |
| 示例：获取600000股票（2015-01-01至2021-01-01）非金融现金流量表财报数据 | | | | | |
| mainclass.get\_CashStatement\_data("600000","2015-01-01","2021-01-01") | | | | | |
| balancesheet | 获取非金融资产负债表财报数据 | code | 股票代码 | 是 |  | str | |
| start | 开始时间 | 是 |  | str | |
| end | 结束时间 | 否 | ‘’ | str | |
| 示例：获取600000股票（2015-01-01至2021-01-01）非金融资产负债表财报数据 | | | | | |
| mainclass.get\_BalanceSheet\_data("600000","2015-01-01","2021-01-01") | | | | | |
| get\_stockIndex\_data | 获取股票指数行情数据 | code | 股票代码 | 是 |  | str | |
| base\_code | 指数代码 | 是 |  | str | |
| start | 开始时间 | 是 |  | str | |
| end | 结束时间 | 否 | ‘’ | str | |
| fqt | 复权值（0不复权，1前复权，2后复权） | 否 | 0 | str | |
| ktype | 数据类型（5五分钟行情，D日行情） | 否 | day | str | |
| bFillNA | 是否为DataFrame为Nan的单元格填充值 | 否 | False | bool | |
| 示例：获取600000股票与000001上证指数（2015-01-01至2021-01-01）日行情前复权数据 | | | | | |
|  | mainclass.get\_stockIndex\_data("600000",”000001”,"2015-01-01","2021-01-01",fqt="1",ktype="D",bFillNA=False) | | | | | |

三、C#类库 & DLL动态库的使用说明

* WtaFinanceFileLibrary.dll 动态库说明

使用.NET FrameWork 4.5框架编写

程序集名称：WtaFinanceFileLibrary

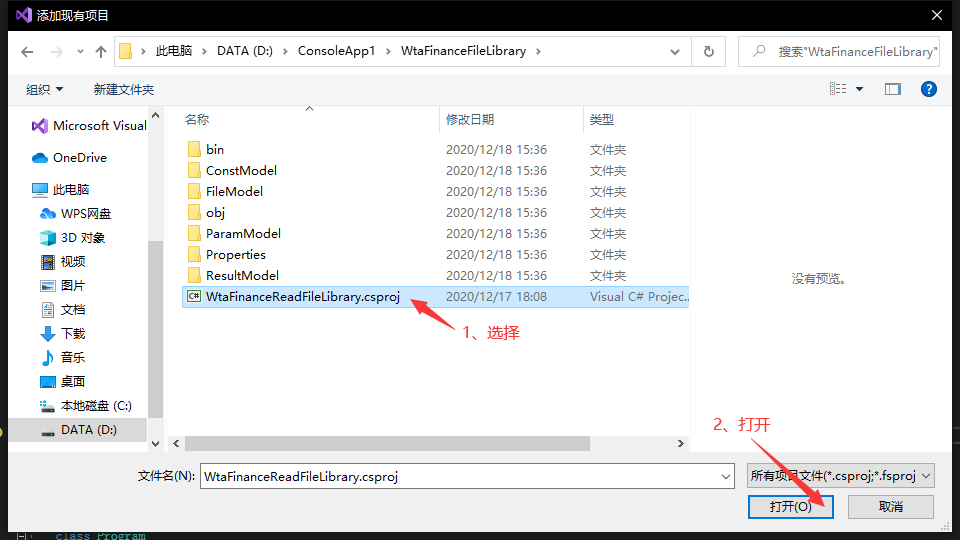
命名空间：WtaFinanceFileLibrary

* 使用方法

1. 类库方式
   1. https://121.42.238.227/svn/wta\_finance/trunk/2 项目实施阶段/2.4 系统开发/2.4.1 系统实现/WtaFinanceC#下获取WtaFinanceFileLibrary 项目拷贝到自己的,工程项目下。



* 1. 鼠标右键解决方案>添加>现有项目>选择拷贝的WtaFinanceReadFileLibrary.csproj工程文件

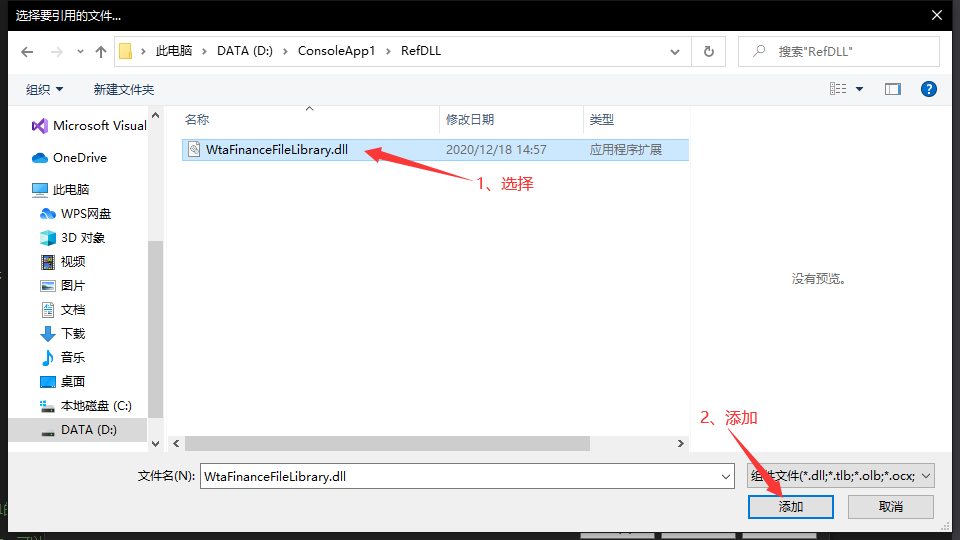


1. DLL动态库使用说明

2.1、 https://121.42.238.227/svn/wta\_finance/trunk/2 项目实施阶段/2.4 系统开发/2.4.1 系统实现/WtaFinanceC#/RefDLL获取WtaFinanceFileLibrary.dll。

2.2、 将WtaFinanceFileLibrary.dll 拷贝到工程项目下

2.3、 鼠标右键引用>添加引用>浏览>找到刚刚拷贝过来的WtaFinanceFileLibrary.dll



* 使用示例

using WtaFinanceFileLibrary;

FileDataSource fileDataSource = new FileDataSource(**Path**);

List<string> codeList = new List<string>(); //股票列表

codeList.Add("000001");

string fqt = "2";//复权值

DateTimeParamModel dateTimeParamModel = null; //时间区段类

List<IndexClass> indexClassList = new List<IndexClass>();

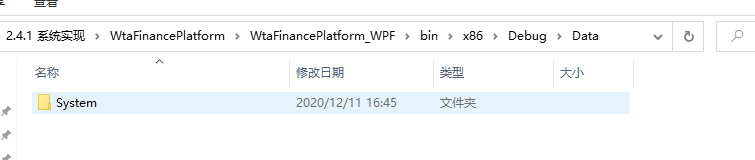
//indexClassList.Add(new IndexClass {Code="1007",FieldList=null }); //查询1007大类时

var **res** = fileDataSource.**GetData**(codeList, fqt, dateTimeParamModel,indexClassList, null);

* **Path**路径说明

生成数据主程序的Data文件夹

例如：



**Path**为绝对路径，参考上图，path的值为：

D:\tby\_svn\wta\_finance\trunk\2 项目实施阶段\2.4 系统开发\2.4.1 系统实现\WtaFinancePlatform\WtaFinancePlatform\_WPF\bin\x86\Debug\Data

* fileDataSource.**GetData**()方法的参数说明

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数** | **说明** | **是否必填** | | **类型** | | |
| codeList | 所查询的股票列表 | 是 | | List<string> | | |
| 示例 | | | | | |
| 示例一： | 列表为空时 | | 所查询的股票 | | |
| 示例二： | codeList.Add("000001");  codeList.Add("000002"); | | 查询000001与000002两只股票数据 | | |
| fqt | 复权值  0：不复权;  1：前复权;  2：后复权; | 是 | | string | | |
|  | 示例 | | | | | |
|  | 示例一： | “0” | | 查询不复权数据 | | |
| dateTime | 时间段 | 否 | | DateTimeParamModel | | |
| DateTimeParamModel包含字段说明 | | | | | |
| **键** | **说明** | **是否必填** | | **格式** | **默认值** |
| StartTime | 开始时间 | 否 | | DateTime | null |
| EndTime | 结束时间 | 否 | | DateTime | null |
| 示例: DateTimeParamModel dateTimeParamModel = new DateTimeParamModel(); | | | | | |
| 示例一： | dateTimeParamModel.StartTime = new DateTime(2020, 1, 1);  dateTimeParamModel.EndTime = new DateTime(2020, 12, 31); | | 获取2020-01-01日至2020-12-31日的数据 | | |
| 示例二： | dateTimeParamModel = null | | 数据无时间限制 | | |
| 示例三： | dateTimeParamModel.StartTime = new DateTime(2020, 1, 1); | | 从2020-01-01 日开始获取的数据 | | |
| 示例四： | dateTimeParamModel.EndTime = new DateTime(2020, 1, 1); | | 获取2020-01-01 日之前的数据 | | |
| System | 获取系统级别数据筛选参数 | 否 | | List<IndexClass> | | |
| IndexClass包含字段说明 | | | | | |
| **键** | **说明** | **是否必填** | | **类型** | **默认值** |
| Code | 大类编码 | 是 | | string | - |
| FieldList | 大类指标集合 | 是 | | List<string> | null |
| 示例: List<IndexClass> IndexClassList = new List<IndexClass>(); | | | | | |
| 示例一： | IndexClassList.Add(new IndexClass { Code = "1007" }); | | | 查询 1007大类所有字段数据 | |
| 示例二： | null | | | 获取所有大类所有指标 | |
| 示例三： | IndexClassList.Add(new IndexClass { Code = 1003" });  IndexClassList.Add(new IndexClass { Code = "1007", FieldList = new List<string>() { "1007002", "1007003",}}); | | | 获取编码为1003的大类所有指标数据  获取编码为1007的大类，指标编码为1007002、1007003的指标 | |

* **res** 返回值说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **说明** |  | | **类型** |
| res | 返回结果 |  | | Dictionary<string, StockResultVM> |
| res 包含字段说明 | | | |
| **键名** | **说明** | | **类型** |
| 股票代码 | 股票代码为键名 | | StockResultVM |
| StockResultVM字段说明 | | | |
| 字段名 | 说明 | 类型 | |
| Data | 大类数据集 | List<ClassResultVM> | |
| SecCode | 股票代码 | string | |
| SecName | 股票名称 | string | |
| Market | 股票市场 | string | |
| ClassResultVM 字段说明 | | | |
| 字段名名 | 说明 | 类型 | |
| Code | 大类的编码 | string | |
| Name | 大类的名称 | string | |
| Total | 数据条目数 | int | |
| FieldList | 获取的字段集合 | List<CodeMap> | |
| Data | 大类数据集 | Dictionary<string, List<string>> | |