

tusharecpp 参考手册

周瑶

2019 年 12 月 24 日

1 安装

依赖项 依赖项主要是 boost 的几个库，如果已经安装了 boost，可以直接使用。

1. boost.beast。C++ 网络库
2. boost.property_tree。用于处理 JSON 数据。
3. boost.tokenizer。分割字符串。

tusharecpp 是一个纯头文件的 C++ 库，将文件夹内的”tusharecpp” 文件夹整个复制到可以让编译器找到的地方就可以使用。接口与 tushare 的 python 接口基本一致。需要编译器支持 C++17 或以上，在 VS 中用时，需要使用 `\std:c++latest`。

2 基本使用

详细的接口参考[tushare 官网](#)。

C++ 代码：

```
#include <iostream>
#include "../tusharecpp/tusharecpp.hpp"
using namespace std;

int main()
{
    auto pro = ts::pro_api("xxxxxxxxxxxxxxxxxxxxxxxx");
    auto data=pro.stock_basic("", "", "");
    cout << "Begin write;" << endl;
    data.to_csv("data.csv");
    std::cout << "Hello World!\n";
}
```

其中 `pro_api` 需要替换成自己的 token，这个 token 要从 tushare 取得。

3 设计原理

代码结构 整个库只由两个文件构成: `data_container.hpp` 和 `tusharecpp.hpp`。每个接口要接受特定的参数, 参数名是直接从 tushare 网站上抓取, 然后使用代码生成的方法产生的, 见于 `code_generate` 文件夹内的 `nb` 文件, 而模板文件是该文件夹内的 `tusharecpp.hpp`。

模板文件内有 `##_AUTO_GENERATED_FIELDS_MAP_##`、`##_AUTO_GENERATED_TYPES_MAP_##`、`##_AUTO_GENERATED_API_##` 几个串, 它们会被替换掉。`fields_map` 中存储的每个接口返回的 `fields`, `types_map` 存储的是返回的每个变量的类型。

数据结构 `data_container` 是用来存储返回的数据的, 内部用 `vector<void*>` 存储每列的指针, 在需要时再 `reinterpret_cast` 回去。因此支持的数据类型有限, 目前有三种: `int`、`string`、`double`。

4 API 说明

这个库在写的时候没有考虑多线程, 也不建议使用多线程, 因为已经具有了爬虫的性质, 如果给服务器的负载过大, 可能有法律上的问题, 再一个也影响其它人使用。