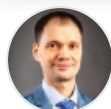


Во время посещения сайта вы соглашаетесь с использованием файлов [cookie](#)

Хорошо



Михаил Шардин ★

личный блог

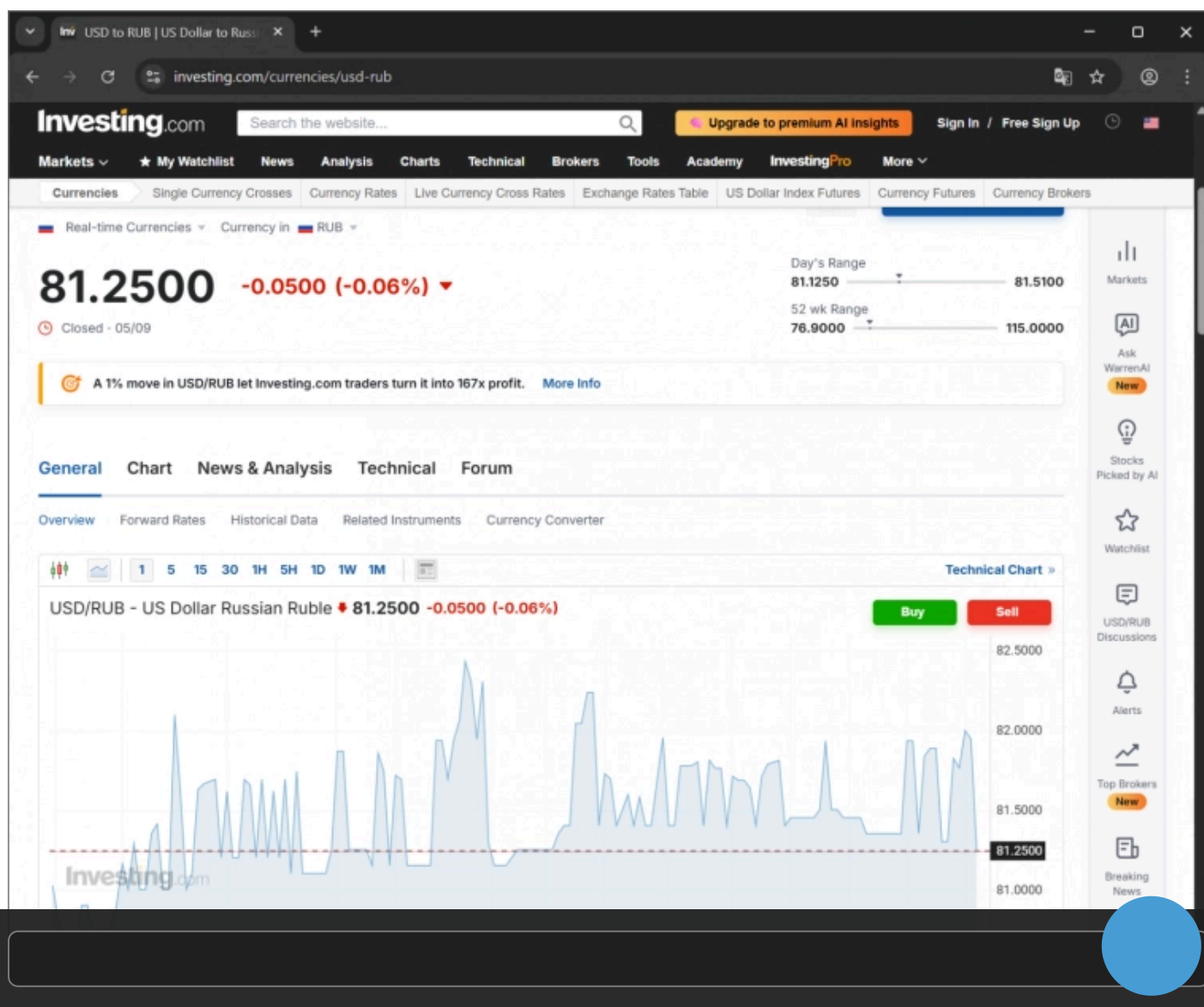


09 сентября 2025, 04:14

+ Подписаться

Автообновляемые котировки в Excel: современный способ брать данные на примере investing.com

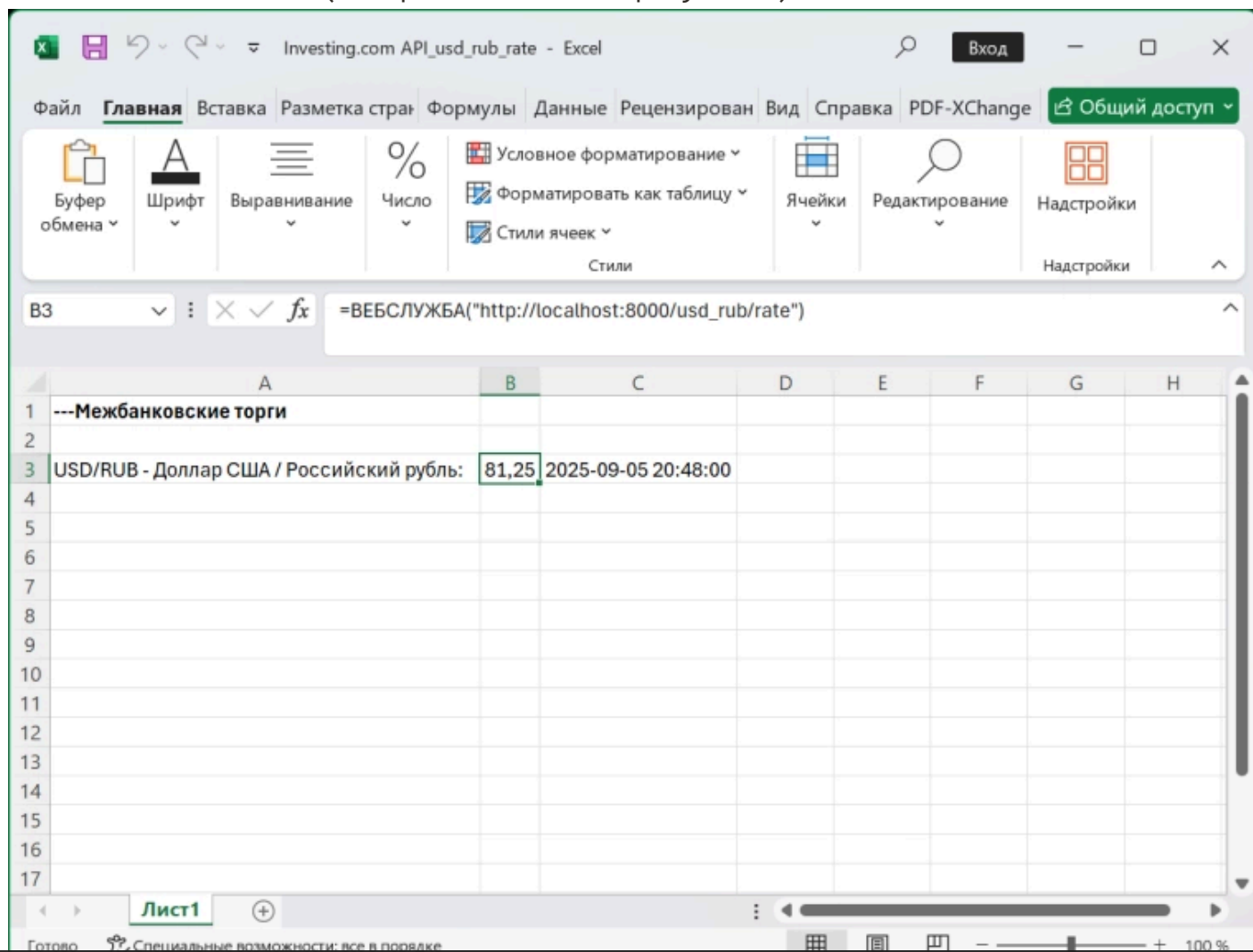
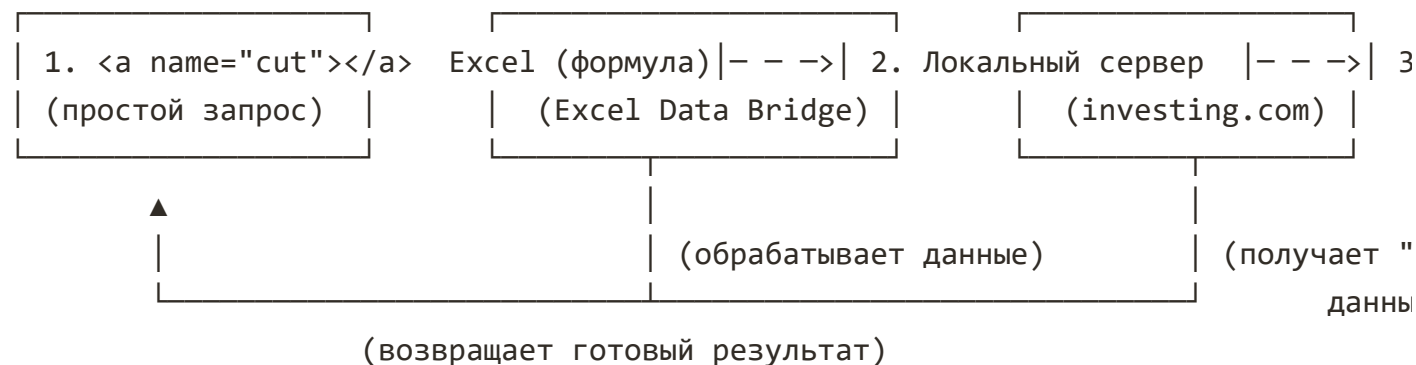
Многие частные инвесторы ведут свои портфели в Excel: это удобно, бесплатно и всё — на вашем компьютере. Но у Excel есть слабое место: он не умеет напрямую «разговаривать» с современными сайтами. Если нужно автоматически подтянуть котировку с конкретной страницы в интернете, встроенные веб-функции часто не справляются: они не умеют обходить современные защиты.



Введите текст комментария

Excel, мы используем на своём компьютере небольшой скрипт-посредник. Excel просто запрашивает у него одно число, а посредник уже «ходит» на сайт, берёт данные, при необходимости обрабатывает их и возвращает в понятном для Excel виде.

Короткая схема работы:

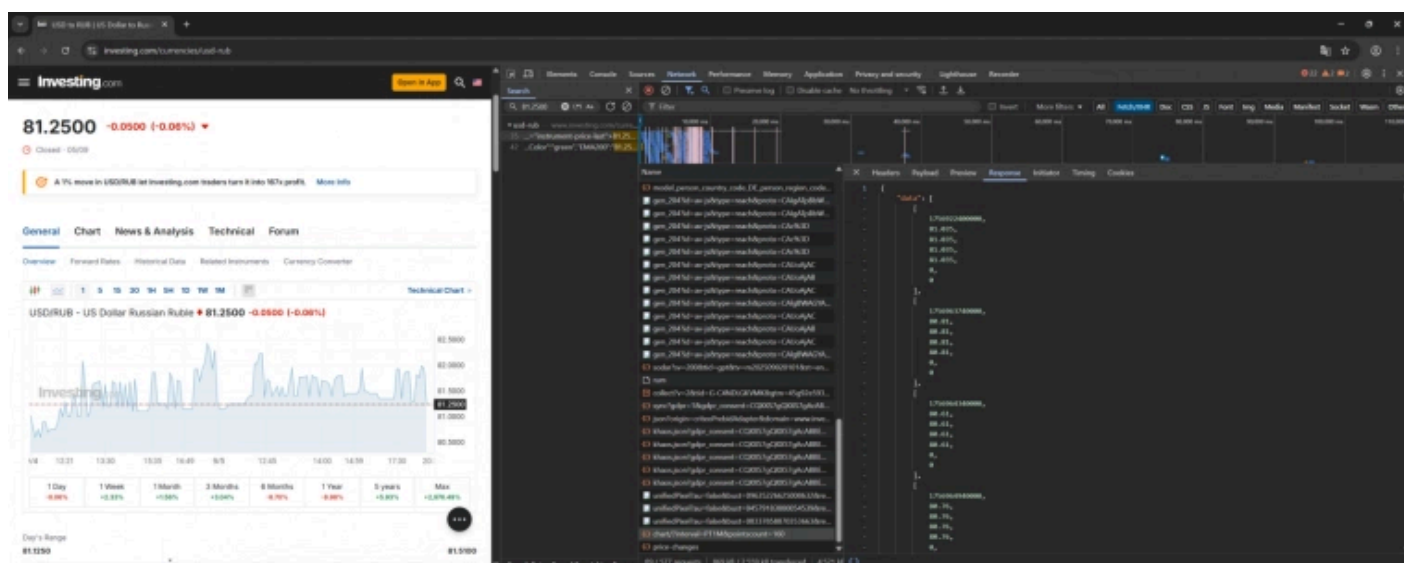


учебный пример. он предназначен исключительно для демонстрации принципа работы с

Все исходные файлы проекта доступны в репозитории на GitHub.

Почему Excel «из коробки» больше не справляется?

Раньше сайты были простыми — статический HTML, и достаточно было послать GET-запрос (когда вы вводите адрес сайта в браузере и нажимаете Enter, ваш браузер отправляет GET-запрос) и прочитать нужный кусок страницы. Сегодня веб — это чаще не страницы, а полноценные приложения: данные подгружаются отдельно через JavaScript, содержимое формируется в браузере и может отсутствовать в исходном HTML. Простая формула Excel этого не видит — она получает «скелет» страницы, а не финальный контент.

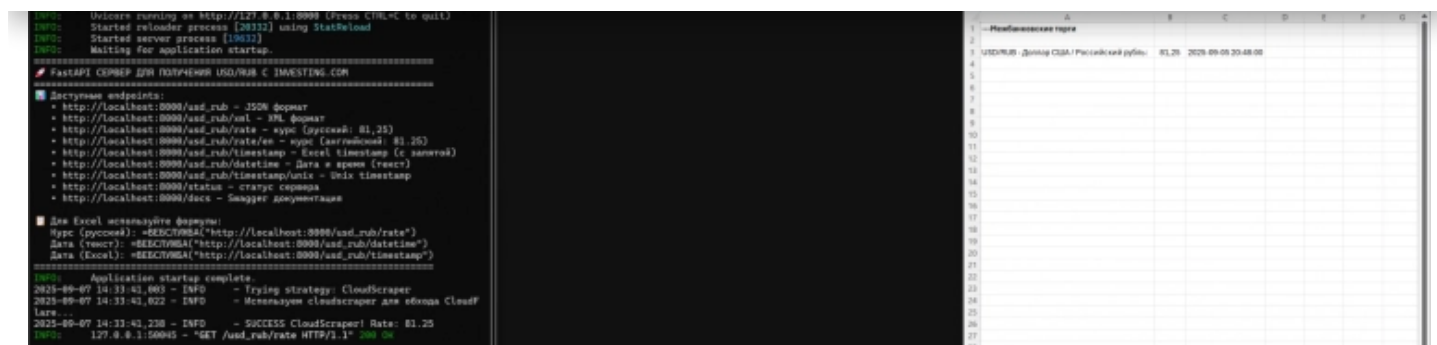


Плюс появились надёжные системы защиты: Cloudflare и их аналоги анализируют трафик и блокируют подозрительные запросы. Запрос из Excel выглядит «механически» — без cookie, без поведенческих отпечатков, без выполнения JS — и его часто сразу отбрасывают или ставят на проверку CAPTCHA.

Нам нужен инструмент, который умеет вести себя как настоящий браузер: выполнять JS, держать сессию, ставить нужные заголовки. Именно таким инструментом станет локальный скрипт-посредник — он «ходит» на сайт как человек/браузер, получает чистые числа и возвращает их Excel в простом виде.

Архитектура нашего решения: строим мост между Excel и вебом

Excel — наш «заказчик». Он делает простой запрос к локальному адресу http://localhost:8000/usd_rub/rate и получает готовое значение. Ему не нужно знать про JavaScript, сессии или CAPTCHA — только чистый текст или XML для ячейки.



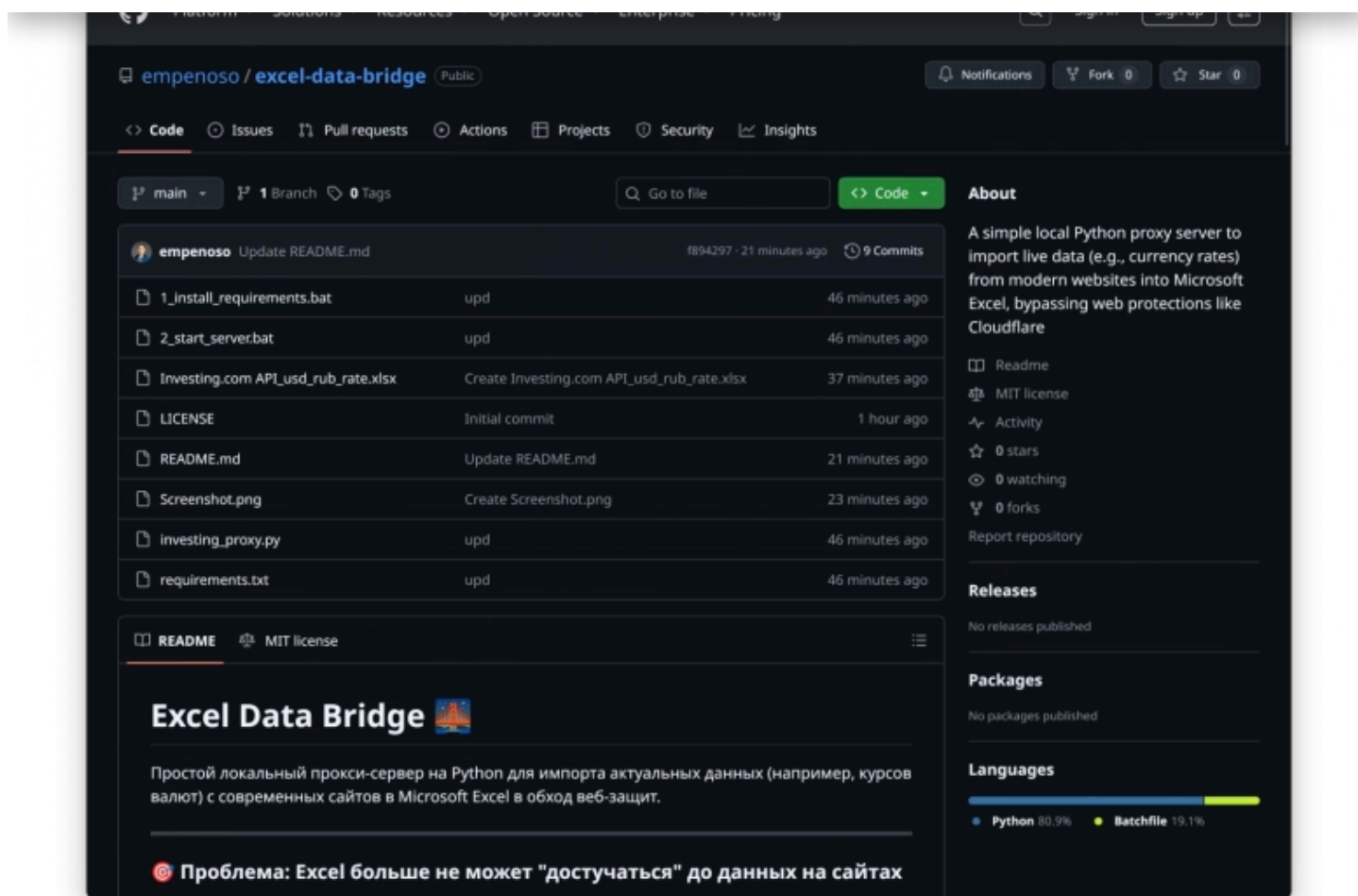
Python + FastAPI — «умный посредник». Лёгкий локальный сервер принимает запрос от Excel, применяет стратегию получения данных, обрабатывает ответ и отдаёт результат в удобном формате. FastAPI даёт быстрый и документированный интерфейс.

requests и cloudscraper — наши «вездеходы». requests надёжен для простых запросов; cloudscraper помогает обходить защиту Cloudflare, имитируя поведение браузера. Сначала пробуем простой запрос, при ошибке переключаемся на cloudscraper и возвращаем то, что Excel «съест».

Итог: прозрачный локальный мост, скрывающий сложности веба и возвращающий котировки в Excel.

Готовим рабочее место и Excel получает данные

Теперь переходим к самому интересному — практической реализации. Наша цель — запустить локальный сервер-посредник и научить Excel обращаться к нему за данными. Следуйте этим шагам, и даже если вы никогда не работали с Python, у вас всё получится.



<https://github.com/empenoso/excel-data-bridge>

Шаг 1: Создание рабочего пространства

Для начала создайте на вашем компьютере отдельную папку, например, excel-data-bridge. В ней мы будем хранить все наши файлы. Это поможет избежать путаницы и обеспечит корректную работу скриптов.

Поместите в эту папку четыре файла, которые были предоставлены ранее:

1. `investing_proxy.py` — наш основной скрипт-посредник.
1. `requirements.txt` — список необходимых Python-библиотек.
1. `1_install_requirements.bat` — установщик зависимостей.
1. `2_start_server.bat` — запускатор нашего локального сервера.

Шаг 2: Установка необходимых компонентов


```

Requirement already satisfied: annotated-types>=0.6.0 in c:\users\михаил\appdata\local\programs\python\python311\lib\site-packages (from pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi->-r requirements.txt (line 1)) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.2 in c:\users\михаил\appdata\local\programs\python\python311\lib\site-packages (from pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi->-r requirements.txt (line 1)) (2.33.2)
Requirement already satisfied: typing-inspection>=0.4.0 in c:\users\михаил\appdata\local\programs\python\python311\lib\site-packages (from pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi->-r requirements.txt (line 1)) (0.4.1)
Requirement already satisfied: anyio<5,>=3.6.2 in c:\users\михаил\appdata\local\programs\python\python311\lib\site-packages (from starlette<0.48.0,>=0.40.0->fastapi->-r requirements.txt (line 1)) (4.10.0)
Requirement already satisfied: idna>=2.8 in c:\users\михаил\appdata\local\programs\python\python311\lib\site-packages (from anyio<5,>=3.6.2->starlette<0.48.0,>=0.40.0->fastapi->-r requirements.txt (line 1)) (3.10)
Requirement already satisfied: sniffio>=1.1 in c:\users\михаил\appdata\local\programs\python\python311\lib\site-packages (from anyio<5,>=3.6.2->starlette<0.48.0,>=0.40.0->fastapi->-r requirements.txt (line 1)) (1.3.1)
Requirement already satisfied: click>=7.0 in c:\users\михаил\appdata\local\programs\python\python311\lib\site-packages (from uvicorn->-r requirements.txt (line 2)) (8.2.1)
Requirement already satisfied: h11>=0.8 in c:\users\михаил\appdata\local\programs\python\python311\lib\site-packages (from uvicorn->-r requirements.txt (line 2)) (0.16.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\михаил\appdata\local\programs\python\python311\lib\site-packages (from requests->-r requirements.txt (line 3)) (3.4.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\михаил\appdata\local\programs\python\python311\lib\site-packages (from requests->-r requirements.txt (line 3)) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\михаил\appdata\local\programs\python\python311\lib\site-packages (from requests->-r requirements.txt (line 3)) (2025.8.3)
Requirement already satisfied: pyparsing>=2.4.7 in c:\users\михаил\appdata\local\programs\python\python311\lib\site-packages (from cloudscraper->-r requirements.txt (line 4)) (3.2.3)
Requirement already satisfied: requests-toolbelt>=0.9.1 in c:\users\михаил\appdata\local\programs\python\python311\lib\site-packages (from cloudscraper->-r requirements.txt (line 4)) (1.0.0)
Requirement already satisfied: colorama in c:\users\михаил\appdata\local\programs\python\python311\lib\site-packages (from click>=7.0->uvicorn->-r requirements.txt (line 2)) (0.4.6)

Проверяем установку uvicorn...
Running uvicorn 0.35.0 with CPython 3.11.9 on Windows
uvicorn установлен успешно!

=====
Установка завершена!
=====
Press any key to continue . . . |

```

Прежде чем наш скрипт сможет работать, ему нужны «помощники» — специальные библиотеки Python. Файл `1_install_requirements.bat` сделает всю работу за вас.

Просто дважды кликните по файлу `1_install_requirements.bat`. Откроется командная строка, где вы увидите процесс установки. Скрипт сначала проверит, установлен ли у вас Python, а затем скачает и установит все библиотеки из файла `requirements.txt`. По завершении вы увидите сообщение «Установка завершена!». Это означает, что всё готово к следующему шагу.

Шаг 3: Запуск локального сервера


```
uvicorn версия: 0.35.0

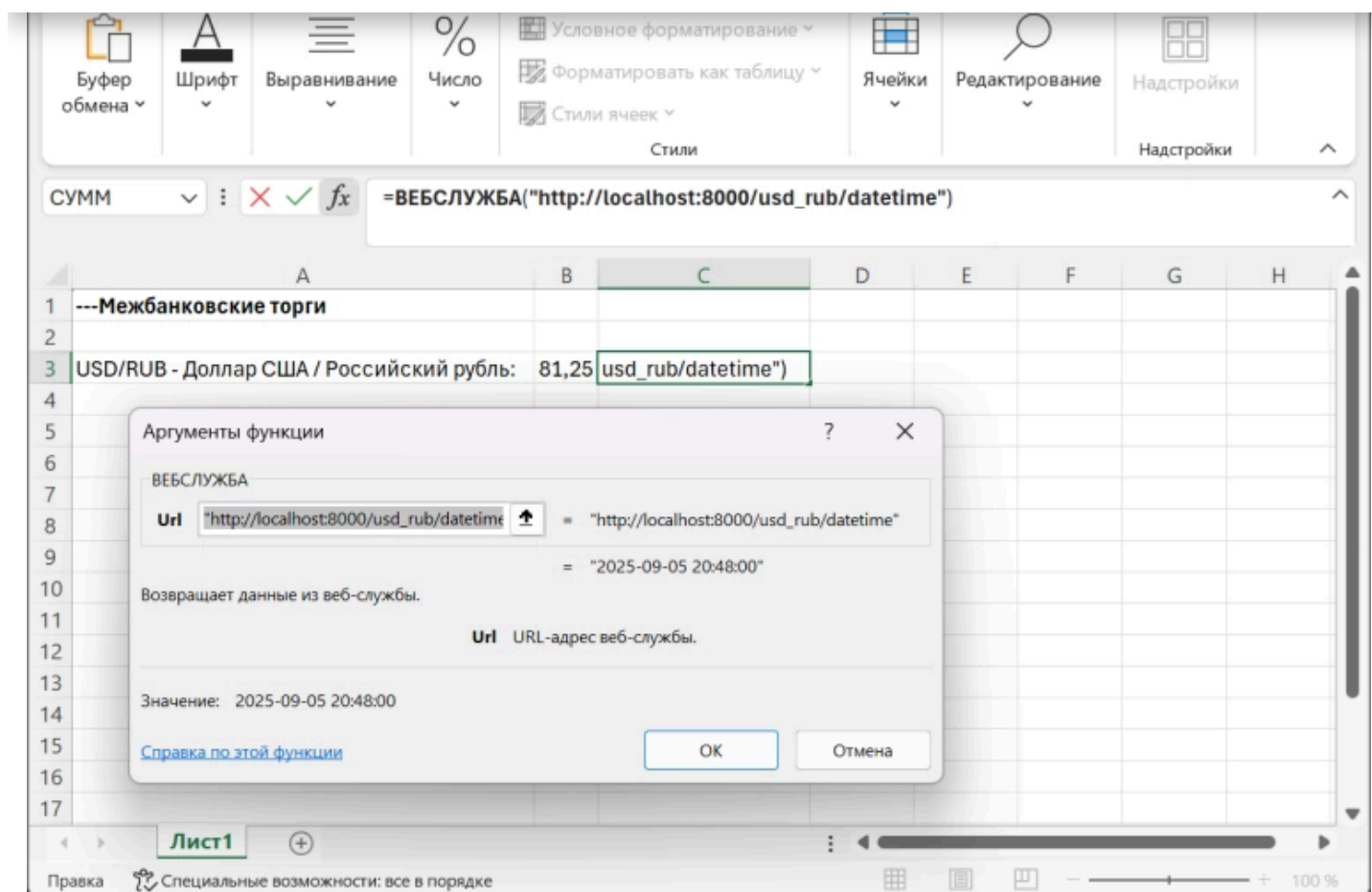
Запуск сервера...
=====
INFO: Will watch for changes in these directories: ['c:\\Users\\Михаил\\Documents\\GitHub\\excel-data-bridge']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [3596] using StatReload
INFO: Started server process [21096]
INFO: Waiting for application startup.
=====
🔥 FastAPI СЕРВЕР ДЛЯ ПОЛУЧЕНИЯ USD/RUB С INVESTING.COM
=====
📄 Доступные endpoints:
• http://localhost:8000/usd_rub - JSON формат
• http://localhost:8000/usd_rub/xml - XML формат
• http://localhost:8000/usd_rub/rate - курс (русский: 81,25)
• http://localhost:8000/usd_rub/rate/en - курс (английский: 81.25)
• http://localhost:8000/usd_rub/timestamp - Excel timestamp (с запятой)
• http://localhost:8000/usd_rub/datetime - Дата и время (текст)
• http://localhost:8000/usd_rub/timestamp/unix - Unix timestamp
• http://localhost:8000/status - статус сервера
• http://localhost:8000/docs - Swagger документация

📄 Для Excel используйте формулы:
Курс (русский): =ВЕБСЛУЖБА("http://localhost:8000/usd_rub/rate")
Дата (текст): =ВЕБСЛУЖБА("http://localhost:8000/usd_rub/datetime")
Дата (Excel): =ВЕБСЛУЖБА("http://localhost:8000/usd_rub/timestamp")
=====
INFO: Application startup complete.
2025-09-07 13:23:11,396 - INFO - Trying strategy: CloudScraper
2025-09-07 13:23:11,421 - INFO - Используем cloudscraper для обхода CloudFlare...
2025-09-07 13:23:11,663 - INFO - SUCCESS CloudScraper! Rate: 81.25
INFO: 127.0.0.1:61414 - "GET /usd_rub/rate HTTP/1.1" 200 OK
2025-09-07 13:23:13,483 - INFO - Trying strategy: CloudScraper
2025-09-07 13:23:13,483 - INFO - Используем cloudscraper для обхода CloudFlare...
2025-09-07 13:23:13,692 - INFO - SUCCESS CloudScraper! Rate: 81.25
INFO: 127.0.0.1:61414 - "GET /usd_rub/datetime HTTP/1.1" 200 OK
```

Теперь, когда все компоненты установлены, запустим наш сервер. Для этого дважды кликните по файлу `2_start_server.bat`.

Снова откроется окно командной строки, но на этот раз оно не закроется. Вы увидите сообщения о запуске сервера, а также список доступных адресов (endpoints), по которым Excel сможет обращаться за данными. Пока это окно открыто, ваш сервер работает и готов принимать запросы от Excel. Если вы закроете это окно, сервер остановится.

Шаг 4: Получение данных в Excel



Откройте Microsoft Excel и выберите любую ячейку. Теперь мы используем встроенную функцию ВЕБСЛУЖБА (WEBSERVICE), которая умеет делать запросы по указанному адресу.

1. **Чтобы получить курс USD/RUB**, введите в ячейку следующую формулу и нажмите Enter: =ВЕБСЛУЖБА("http://localhost:8000/usd_rub/rate")

1. **Чтобы получить дату и время котировки**, введите в соседнюю ячейку: =ВЕБСЛУЖБА("http://localhost:8000/usd_rub/datetime")

Excel отправит запрос на ваш локальный сервер, тот, в свою очередь, сходит на investing.com, получит данные и вернёт их в ячейку.

81,25

Это просто пример — как можно модифицировать под себя?

Это лишь базовый пример, а не готовый универсальный инструмент. Скрипт показывает принцип: Excel делает простой запрос, а посредник достаёт данные с сайта и возвращает результат. Но у каждого инвестора свои задачи: кому-то нужны котировки акций, кому-то — нефть или золото, кто-то захочет загружать таблицы. Именно поэтому код придётся адаптировать под конкретный сайт, формат ответа и даже частоту обновлений. Главное — вы держите в руках рабочий шаблон, который легко модифицировать под себя.

Заключение

Мы не просто решили локальную задачу получения котировок — мы освоили мощный подход к интеграции Excel с современным вебом. Создав локальный API-посредник, мы научили старый добрый Excel говорить на языке современных веб-приложений, обходя их защиты и получая актуальные данные.

Этот мост между Python и Excel открывает широкие возможности для автоматизации рутинных операций и делает ваш инвестиционный портфель по-настоящему «живым» — с автообновляющимися котировками, курсами валют и любыми финансовыми данными из интернета.

Автор: Михаил Шардин

 [Моя онлайновая визитка](#)

Дмитрий Мамедиев и трейдер Smart Lab Conf 2025, который пройдет 25 октября в Москве.

На выступлении 25 октября в зале №7 (спекуляции) в 12:00 я подробнее разберу технические детали, покажу живые примеры и отвечу на ваши вопросы.

Буду рад видеть всех, кто интересуется автоматизацией и новыми подходами в работе с финансовыми данными.

До встречи!

Конференции смартлаба

Smart Lab

Excel

торговые роботы

11.2K 

☆ 51

💬 48

❤️ 66



Михаил Шардин

📍 Пермь

👤 398 📊 4 228

📅 с 23 января 2019

🔗 +HreHDn1F5CzjN...

+ Подписаться

48 КОММЕНТАРИЕВ

Сначала старые ▾



Whalerman

09 сентября 2025, 07:40

Михаил, спасибо! Хорошие и полезные посты!

— Показать 1 ответ

↩️ 👍 +1 💬



Дмитрий Мамедиев

09 сентября 2025, 07:45

вообще то можно. создать подключение , вставляем ссылку и всё. берём от сюда <https://iss.moex.com/iss/reference/>

— Показать 7 ответов

↩️ 👍 💬

Ждём новую версию экселя где это пофиксят.

— Показать 6 ответов



DrManhattan

09 сентября 2025, 12:48



„Мы не просто решили локальную задачу получения котировок “

А сколько вас там в банде?

— Показать 1 ответ



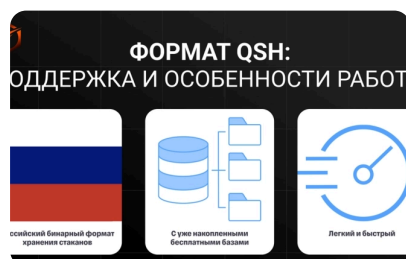
Ещё 8 комментариев

Напишите комментарий...



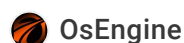
ОТПРАВИТЬ

Читайте на SMART-LAB:



Формат QSH в OsEngine: поддержка и особенности работы

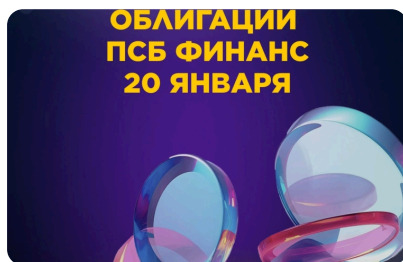
Недавно OsEngine начал поддержку бинарного формата хранения и трансляции данных по стаканам. Это было нужно, чтобы:...



18:01

 Softline

17:16

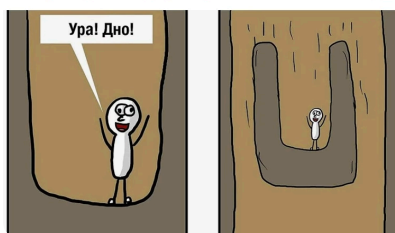
**⚡ Объявляем условия нового размещения**

20 января финтех-сервис ПСБ Финанс (бренд CarMoney) начнет размещение нового выпуска облигаций с плавающим купоном...

 CarMoney

17:52

кабре идет ко дну - хуже не было никогда


**Хедхантер. Ситуация на рынке труда в декабре идет ко дну - хуже не было никогда**

Вышла статистика рынка труда за декабрь 2025 года, которую Хедхантер публикует ежемесячно, что же там интересного:...

 Mozgovik

13.01.2026

Установите приложение Смартлаба:

 RuStore AppGallery App Store[О смартлабе](#)[Реклама](#)[Полная версия](#)

Московская Биржа является спонсором ресурса smart-lab.ru
Источник: ПАО Московская Биржа