empenoso
9 сен 2025 в 05:23

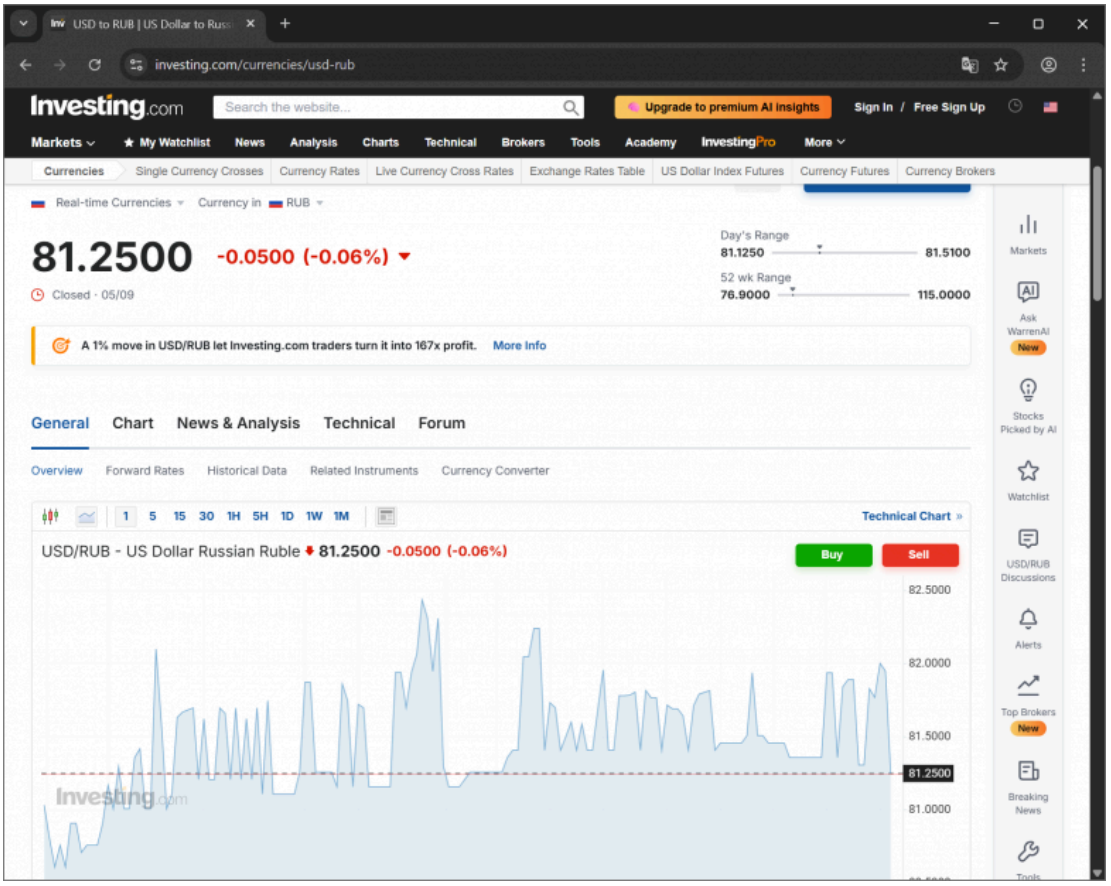
Как получать котировки с любых сайтов в Эксель на примере investing.com

Простой 5 мин 12K

Open source*, Python*, Финансы в IT, Проектирование API*

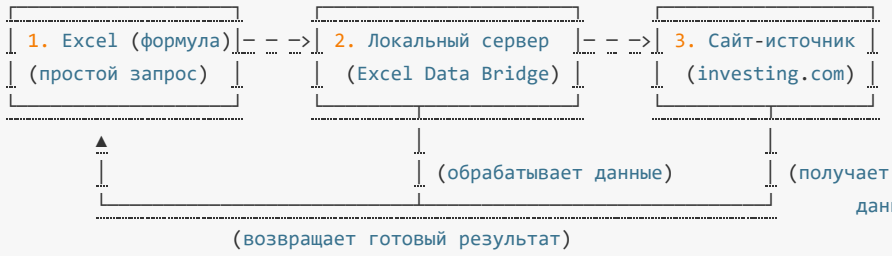
Кейс

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.

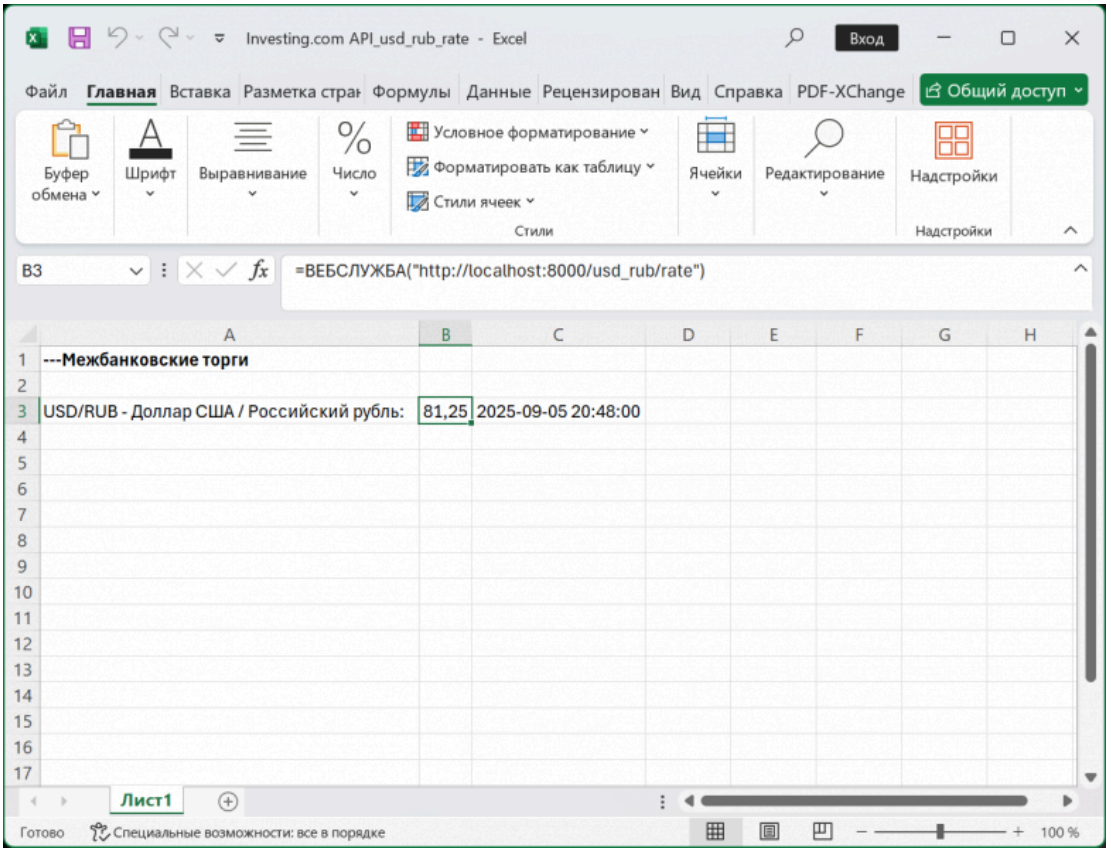


В этой статье я покажу простой и надёжный способ заставить Excel получать котировки практически с любого сайта — на примере курса USD/RUB с investing.com. Идея не требует глубоких технических знаний: вместо того чтобы пытаться что-то делать со страницей в Excel, мы используем на своём компьютере небольшой 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:



Объяснить код с SourceCraft



Приведённый далее Python-скрипт (набор инструкций для этого «посредника») — это учебный пример: он предназначен исключительно для демонстрации принципа работы с API и веб-технологиями. Я не призываю и не рекомендую использовать его для обхода правил каких-либо сайтов.

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.
2. `requirements.txt` — список необходимых Python-библиотек.
3. `1_install_requirements.bat` — установщик зависимостей.
4. `2_start_server.bat` — запускатор нашего локального сервера.

Шаг 2: Установка необходимых компонентов

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

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 ")`
2. **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.

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 онлайн-визитка
📢 Telegram «Умный Дом Инвестора»

9 сентября 2025

Теги: investing.com, api, excel

Хабы: Open source, Python, Финансы в IT, Проектирование API

Комментарии 10

Публикации

ЛУЧШИЕ ЗА СУТКИ ПОХОЖИЕ

DrArgentum
7 часов назад

Ненормальные непотребства, трюки, хаки и алгоритмы на C

Простой 10 мин 3.7K

Обзор

+25 31 9

Sivchenko_translate
6 часов назад

std::move ничего никуда не двигает: подробный рассказ о категориях значений в C++

35 мин 3.4K

Перевод

+21 34 2

TrexSelectel
6 часов назад

Полезные ресурсы для тестировщиков: подборка от специалистов Selectel

3 мин 2.7K

Мнение

+21 7 0

shiru8bit
6 часов назад

Игра во время загрузки игры

Простой 15 мин 3K

Ретроспектива

+19 3 2

CreatorLAB
22 часа назад

Конфигуратор микроконтроллеров STM8S103/105

10 мин 10K

+18 39 8

dronnix
8 часов назад

Black-White Array: новая структура данных с O(log N) аллокаций

Средний 8 мин 5.1K

Обзор

+17 39 2

beget_com
8 часов назад

Автомобили-конструкторы, кафе с удалёнными официантами и отстреливающиеся батареи: 15 проектов промдизайна 2025

6 мин 5.2K

Кейс

+16 2 1

Grishandin
8 часов назад

Закономерности в данных вместо догадок: как мы помогаем студентам дойти до конца курса

Средний 8 мин 3.9K

Кейс

+16 5 0

sergbe
8 часов назад

Рецензия на книгу «Принципы модернизации программных архитектур»

Простой 9 мин 4.8K

Мнение

+14 7 0

stasvetokhin
9 часов назад

Как выбрать идею для инди-игры и не потратить годы впустую

Простой 8 мин 5K

Кейс

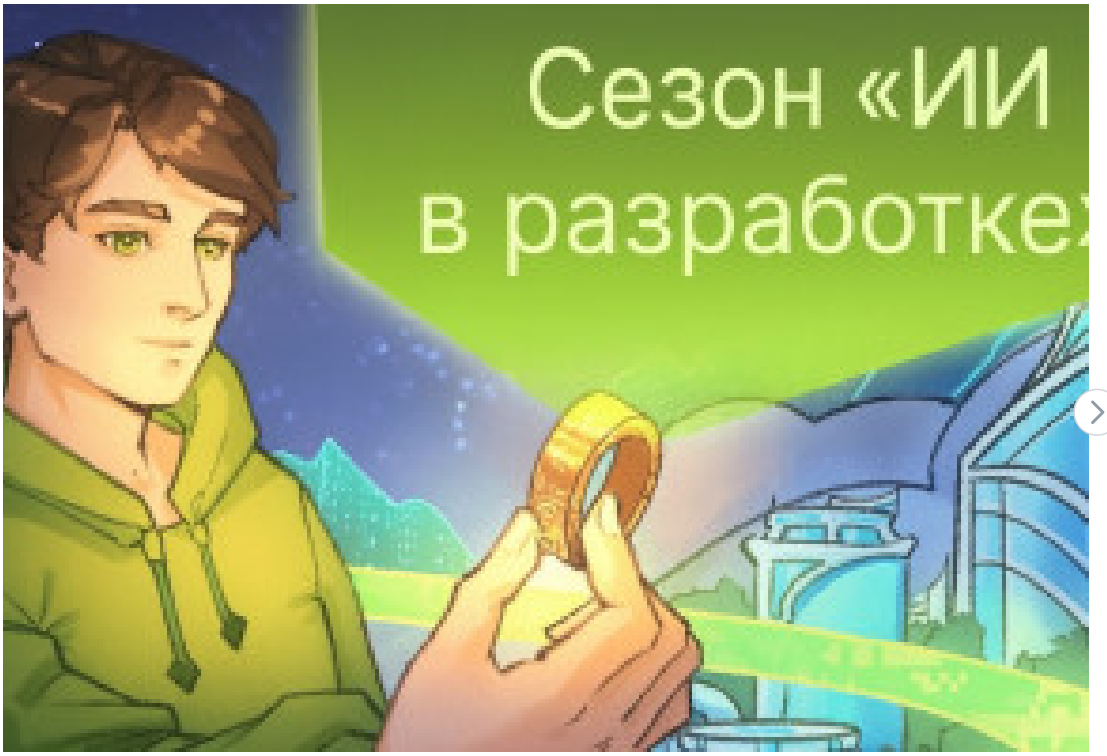
+13 10 2

Подход к облаку, где сложность остаётся на стороне провайдера

Турбо

Показать еще

БЛИЖАЙШИЕ СОБЫТИЯ



25 ноября 2025 – 16 января 2026

Сезон «ИИ в разработке»

Онлайн

Разработка Другое

Больше событий в календаре