



Горячее Лучшее Свежее ...



empenoso Программирование на python

FinLabPy: когда брокеры отказались создавать стандарт — его создал один человек

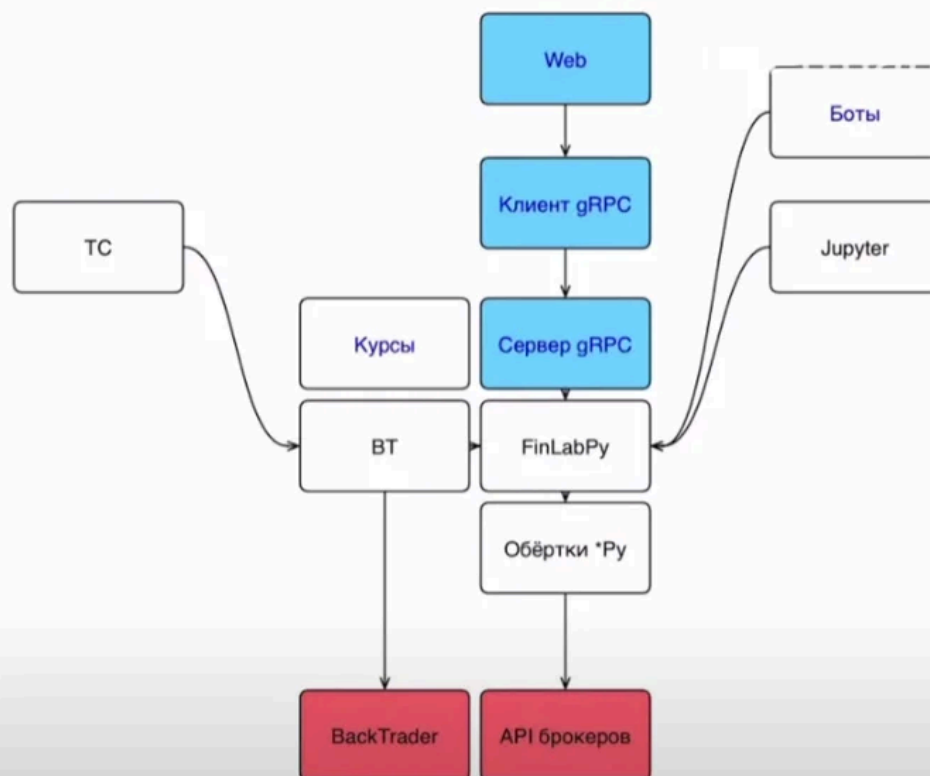
🕒 1 месяц назад 👁 4.1K

В последнее время я активно занимаюсь автоматизацией торговли и знакомлюсь с разными решениями, два раза летал на конференции, познакомился с интересными людьми. На этом фоне я наткнулся на open-source проект [cia76/FinLabPy](#), о котором я уже давно слышал, но никогда не разбирался подробно.

Российская алготорговля переживает странный период: возможности растут, но стандартизации как будто не существует. Брокеры выпускают свои API, но каждый из них живёт в отдельной вселенной — со своим обозначением тикеров, задержками и внезапными отключениями.

Про проблемы алготорговли на Московской бирже почти не пишут, хотя есть мнение, что 60% оборота биржи создаётся роботами. А вот [автор этого проекта Игорь Чечет](#) на своём вебинаре рассказывает о том с какими проблемами может столкнуться частный инвестор, когда приходит в алгоритмическую среду.

Начну с главного - какую вообще проблему решает FinLabPy?



Что такое FinLabPy и какую проблему он решает

Используйте аккаунт Яндекса для входа на сервис

Безопасный вход без дополнительной регистрации на сайте

Войти с Яндекс ID

[Забыли пароль?](#)

или продолжите с

Войти с Яндекс ID

Войти через VK ID

- Промокоды
- Работа
- Курсы
- Реклама
- Игры
- Пополнение Steam

cia76/FinLabPy — это унифицированная платформа для анализа рынков, прототипирования торговых идей, тестирования стратегий и запуска автоторговли через нескольких российских брокеров.

Необходимость создания такой библиотеки возникла потому что российские брокеры реализовали API «каждый в меру своих возможностей».

Несколько примеров:

- **Финам:** может самостоятельно отваливать подписки.
- **Т-Инвест:** присылает сделки пачками и с задержкой; бары иногда запаздывают на 2–3 минуты.
- **Алор:** любит перезагружать сервер прямо во время торгов.
- **QUIK:** использует свою экзотическую систему тикеров вида TQBR.SBER.

У каждого своя спецификация данных, свои ограничения и свои «фишки», которые на деле превращаются в головную боль.

FinLabPy по словам его создателя забирает всю эту боль себе: переподключения, нормализация данных, логирование, кэширование, обработка ошибок, единые тикеры — всё это зашито в open-source библиотеке.

Архитектура: три уровня, которые упрощают жизнь разработчику

FinLabPy устроена на трёх уровнях что меня удивило:

1. Нижний слой: нативные API брокеров.

REST, WebSocket, GRPC — всё, что брокер даёт.

2. Средний слой: Python-обёртки.

Отдельные проекты под каждого брокера:

- AlorPy
- FinamPy
- QuikPy
- TinvestPy (Т-Инвест)
- (в работе) обёртка для БКС

3. Верхний слой: FinLabPy

Единый интерфейс, единая модель данных, единая логика. При этом доступ к «уникальным» функциям конкретного брокера сохраняется:

FinLabPy → provider.provider → уникальные методы обёртки.

Вообще меня порадовало, что внутри нет самодельных велосипедов

Технологический стек FinLabPy

Автор сознательно взял лучшие решения рынка и встроил их в экосистему. Никакого изобретения велосипеда.

Основные элементы:

- Python
- Backtrader — тестирование и автоторговля



Пикабу Игры
+1000 бесплатных
онлайн игр



Битва Героев

Ролевые, Приключения,
Мидкорные

Играть



→ национальныепроекты.рф

Экспорт хвой из Пермского края вырос почти в три раза.

Читайте, как нацпроект помогает экспортировать российские товары в новых условиях.

Узнать больше

Топ прошлой недели

- Animalrescued
41 пост
- a.lisik
3 поста
- Oskanov
8 постов

Посмотреть весь топ



**Лучшие посты
недели**



- шаг цены,
- минимальный лот,
- доступная история свечей (у разных брокеров — разная),
- прочие параметры.

Есть отдельный модуль для работы с расписаниями Московской биржи.

Стратегия всегда знает, когда открытие/заккрытие сессии, какая сейчас стадия торгов, и может корректно обрабатывать данные.

Telegram-интеграции и мультиброкерская работа

Меня, работающего с несколькими брокерами и, особенно это зацепило. У автора всё это сведено в слой — «мультиброкерский пакет», где есть готовые Telegram-боты:

- сводка по всем счетам,
- баланс,
- открытые позиции,
- уведомления.

Используется aiogram, поддерживается и Discord.

Зачем мультиброкерность?

- запуск одной стратегии на нескольких брокерах;
- диверсификация инфраструктурных рисков;
- удобство тестирования;
- возможность легко «мигрировать» между брокерами.

Переключение брокера = изменение параметра в конфиге.

Backtrader: стандарт де-факто, но с оговорками

Автор прямо говорит, что Backtrader — мощный инструмент, но он заброшен.

В планах:

- либо создать собственный форк и привести архитектуру в порядок,
- либо полностью переписать движок,
- но сохранить совместимость со всеми существующими стратегиями.

До создания шаблона я тоже пытался разработать собственный GUI для Backtrader — простой интерфейс. Но проект особо не взлетел, хотя это и была попытка создать удобный шаблон, который каждый сможет расширять под себя.

Веб-интерфейс, gRPC и клиент-серверная архитектура

Логическим продолжением разговора стала архитектура будущей версии.

Автор явно движется к полноценной платформе. Планы развития со слов Игоря выглядят так:

Веб-интерфейс. Для визуализации, анализа и, возможно, полноценной работы с роботами. Примерно как «домашний терминал».

gRPC. FinLabPy + роботы работают на VPS (сервер), а аналитика и управление — с локального ПК через gRPC-клиент. Это даст:

- безопасность,
- скорость,
- возможность разнести вычисления и интерфейс.

Из трёх часов стрима я отметил для себя несколько рекомендаций.

Практические советы разработчикам

Игорь на стриме регулярно подчёркивает несколько вещей:

1. Не изобретайте велосипеды.

TradingView для графиков, TA-Lib для индикаторов, Backtrader для торговли — всё уже существует.

2. Запускайте роботов только на VPS под Linux.

3. **Используйте подробное логирование (DEBUG).** Особенно в период отладки подписок и торговых операций.

4. Помните: унификация ограничена возможностями «самого слабого» брокера.

Если нужны уникальные функции — используйте методы конкретной обёртки напрямую.

Немного юридических моментов

Не обошлось и без юридических деталей — неожиданный, но важный момент, который автор пояснил. Автор планирует сертифицировать FinLabPy в реестре российского ПО Минцифры.

Это нужно:

1. для защиты авторских прав;
2. для потенциальной интеграции библиотеки брокерами или биржей.

Но для обычных трейдеров это вроде как никак не меняет ситуацию: проект планируется open-source.

Что пока остаётся «за кадром»

Как человек, который интересуется не только инфраструктурой, но и моделями, я отметил список тем, которые автор пока не стал раскрывать:

- продвинутые методы бэктестинга (walk-forward, Монте-Карло),
- управление рисками и портфельные модели,
- оптимизацию производительности,
- структуру конфигов,
- интеграцию LLM / AI в торговые системы.

Фокус шёл именно на инфраструктуру и унификацию.

Итог

Лично я воспринимаю [cia76/FinLabPy](#) как один из самых многообещающих open-source проектов под российскую алготорговлю. Это попытка создать единый стандарт, которого брокеры не смогли (или не захотели) создать сами.

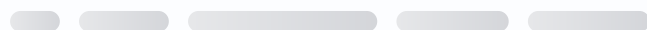
Если вы пишете торговых роботов под российских брокеров — присмотреться к FinLabPy определённо стоит.

Автор: Михаил Шардин

[Моя онлайн-визитка](#)

[Telegram «Умный Дом Инвестора»](#)

18 ноября 2025



РЕКЛАМА



[aliexpress.ru](#)

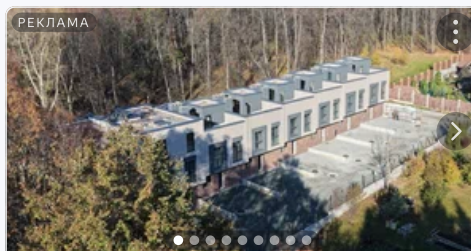
**Мини-сервер MaixPy NanoKVM
PiKVM для вашего проекта**

4 710 ₽ **-17%** 5 674 ₽

[Уже в корзине](#)

[Узнать больше](#)

РЕКЛАМА



[tiliadom.ru](#)

**Таунхаусы Tilia в липовой
роще. 15 минут до центра
Перми**

[Узнать больше](#)



Программирование на python

942 поста • 12K подписчика

[Добавить пост](#)

[Подписаться](#)



Правила сообщества

Публиковать могут пользователи с любым рейтингом. Однако!

Приветствуется:...

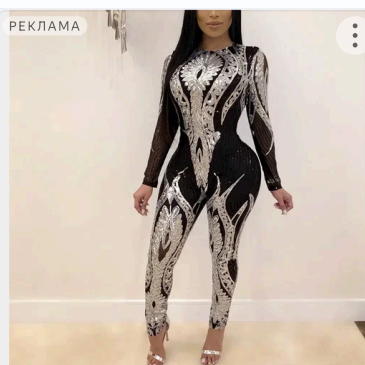
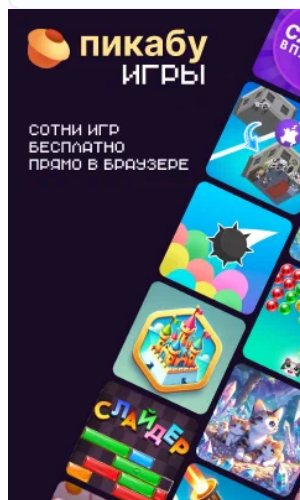
[Подробнее](#) ✓

[Все комментарии](#)

[Автора](#)

Раскрыть 11 комментариев

Чтобы оставить комментарий, необходимо [зарегистрироваться](#) или [войти](#)

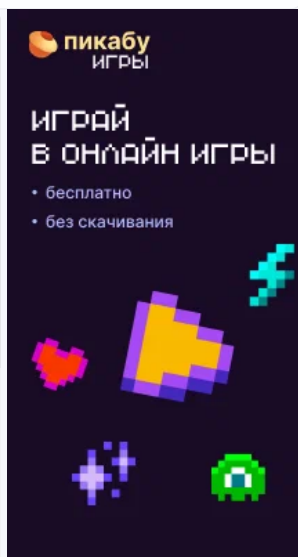


РЕКЛАМА
aliexpress.ru

Женский прозрачный комбинезон Matteobenni

2 044 ₽

[Узнать больше](#)



● — ■ —

● — ■ —

● — ■ —

