

API и ДОСТУП К ДАННЫМ

Краткие теоретические материалы

1 Что такое API

- **API (Application Programming Interface)** - интерфейс, через который программы взаимодействуют друг с другом. При сборе данных API позволяет получать информацию сразу в структурированном виде, чаще всего в формате JSON. При парсинге HTML-страниц данные приходится извлекать из верстки с помощью селекторов или регулярных выражений.
- **HTTP-методы.** Чаще всего используются методы GET и POST. GET применяется для получения данных, а параметры обычно передаются в URL. POST используется для отправки данных или сложных запросов. Параметры и полезная нагрузка (payload) в этом случае передаются в теле запроса.
- **Авторизация.** Для доступа к защищенным эндпоинтам используются токены, например `access_token` или `api_key`. Токены нельзя добавлять в публичные репозитории, поскольку они могут давать доступ к аккаунту. Обычно их хранят в переменных окружения или локальных конфигурационных файлах, добавленных в `.gitignore`.

2 Telegram (MTPROTO API)

Для сбора данных из Telegram-каналов обычно используется библиотека `telethon`, которая работает с MTPROTO API. Последовательность действий выглядит так:

- зарегистрировать приложение на сайте `my.telegram.org` и получить `api_id` и `api_hash`;
- создать асинхронный клиент `TelegramClient` и пройти первичную авторизацию;
- получить сущность (`entity`) выбранного публичного канала или чата;
- загрузить историю сообщений с помощью асинхронного итератора `iter_messages`, конструкции `async for` и библиотеки `asyncio`;
- извлечь из сообщений дату, текст, число просмотров, пересылки и другую доступную информацию для дальнейшего анализа.

3 ВКонтакте (REST API)

VK предоставляет REST API для доступа к данным сообществ, пользователей и медиа. Основные элементы работы с ним:

- получение ключа доступа - сервисного или пользовательского токена;
- использование метода `wall.get` для загрузки постов со стены сообщества. Основные параметры - `domain`, `count` и `offset`;
- загрузка более 100 постов с помощью пагинации. На каждой итерации значение `offset` увеличивается;

- обработка ответа в формате JSON. Из ответа можно получить текст и дату поста, а также число лайков, комментариев и репостов. Дата возвращается в формате Unix timestamp и преобразуется в объект `datetime`.

4 Московская биржа (MOEX ISS)

Информационный сервер MOEX ISS (Informational & Statistical Server) предоставляет исторические котировки, свечи и справочники инструментов без авторизации.

- Данные доступны через GET-запросы к эндпоинтам вида `iss.moex.com/iss/history/...` или `iss.moex.com/iss/engines/....`
- **Пример запроса** дневных свечей по тикеру SBER:
`https://iss.moex.com/iss/engines/stock/markets/shares/boards/TQBR/securities/SBER/candles.json?from=2024-01-01&till=2024-01-31&interval=24`
- За один запрос API возвращает не более 100 торговых дней. Для загрузки данных за длительный период параметр `from` последовательно сдвигается.
- Ответ можно получить в формате JSON или CSV. CSV удобно сразу загружать в `pandas.DataFrame` с помощью функции `read_csv`.

5 Hidden API и DevTools

Многие современные сайты не публикуют документацию API, но используют внутренние запросы для динамической загрузки данных. Такие запросы, например, отправляются при прокрутке ленты или переключении вкладок без перезагрузки страницы.

- Откройте инструменты разработчика (DevTools, клавиша F12) и перейдите на вкладку **Network**.
- Отфильтруйте запросы по типу **Fetch/XHR** и посмотрите, какие GET- или POST-запросы браузер отправляет при работе со страницей.
- Если запрос возвращает структурированные данные в формате JSON, скопируйте его URL, заголовки (Headers) и параметры (Payload).
- Воспроизведите запрос с помощью библиотеки `requests` в Python. При необходимости передайте заголовки `User-Agent`, `Cookie` или `Authorization`, которые использует браузер.

6 OpenAI API и формат ChatML

OpenAI API позволяет отправлять текстовые запросы большим языковым моделям (LLM). В формате **ChatML (Chat Markup Language)** запрос представлен упорядоченным списком сообщений `messages`. Каждому сообщению соответствует определенная роль:

- **system** - системный промпт, который задает общее поведение, стиль и ограничения модели, например требование отвечать в формате JSON;

- **user** - запрос, вопрос или контекст от пользователя;
- **assistant** - предыдущие ответы модели, которые поддерживают контекст диалога.

Запрос отправляется методом POST на эндпоинт `/v1/chat/completions` с заголовком авторизации **Authorization: Bearer <API_KEY>**. Ответ модели можно использовать для классификации, суммаризации, извлечения сущностей или генерации синтетических данных.