

API

API — это набор общедоступных ручек/рычагов приложения, которые позволяют взаимодействовать с приложением. Часто приложения бывают очень большие, сложные и запутанные для рядового пользователя. API представляет из себя простой и понятный пользователю (чаще всего этот пользователь — другой разработчик) набор инструментов для работы с приложением.

Перед тем, как перейти к примерам из мира программирования, давайте обратимся к примеру-анalogии из реальной жизни: мы приходим в ресторан и просим официанта подать нам коктейль и закуски. Официант в свою очередь передает наш заказ на кухню, где происходит приготовление блюд и напитков. Официант для нас является лишь посредником для удобной работы с кухней и поварами: мы можем не знать поваров, не знать жаргонных словечек и сокращенных названий блюд, не знать, на каком языке они разговаривают. За нас всю эту работу проделает официант: он знает все сокращенные названия блюд, знает кому отдать заказ на закуски (повару), а кому — на напиток (бармену). Как итог, через несколько минут официант приносит желаемый заказ, а клиент остается доволен.

Этот пример довольно хорошо описывает работу API и бэкенда в целом:

- Языки, на которых разговаривают повара и бармен — это языки программирования
- Приготовление блюд можно сравнить с обращением к базе данных овощей и фруктов и обработке этих данных
- Официант также, как и мы, может не знать, что на самом деле творится на кухне и в каких пропорциях бармен смешивает алкоголь, но тем не менее он знает как к ним обратиться и как донести до них наше желание

Как вы могли догадаться, официант — это и есть API.

Давайте наконец дадим расшифровку этому термину. API — application programming interface или программный интерфейс приложения — это описание способов обращения с нашим приложением на понятном клиенту (пользователю) языке.

Например, мы можем сконструировать невероятной сложности и проработанности приложение по поиску отелей, но для клиента мы даем лишь один публичный адрес для обращения: `site.com/hotels`. Здесь `/hotels` — это самый маленький элемент нашего API, называемый эндпоинтом или "ручкой", за которую можно дернуть и получить список отелей. Написанием таких ручек мы будем заниматься в этом курсе.

API — это не только про веб-приложения

API может быть как у веб-приложений (мы сделаем его в этом курсе), так и, например, у библиотек в Python. Если вы захотите отправить HTTP запрос на какой-либо сайт, вы можете воспользоваться библиотекой `requests` и отправить запрос в одну строчку

через `requests.get("https://ya.ru").text`. Здесь `get` — это публичный метод библиотеки, который авторы предлагают использовать пользователю для отправки GET запросов. Всего 1 действие и у вас перед глазами HTML код главной страницы Яндекса. Под капотом же происходят гораздо более сложные операции, которые вряд ли интересны рядовому разработчику: добавление заголовков к запросу, работа с редиректами, отправка запроса и др.



FastAPI

FastAPI — это фреймворк (с технической точки зрения это библиотека) на языке Python, который помогает нам быстро и красиво писать эндпоинты (здесь и далее будет использоваться слово эндпоинт вместо "ручки").

FastAPI вышел в свет совсем недавно — в 2018 году. Его написал 1 человек и продолжает совершенствоваться по сей день. FastAPI на текущий момент занимает 3е место по популярности среди Python backend фреймворков, уступая Django и Flask. Тем не менее, на глазах автора прослеживается явная тенденция в сторону перехода на

FastAPI: появляется все больше вакансий, где требуют знать как один из фреймворков Django/Flask/FastAPI, так и только FastAPI.

Фреймворк завоевал любовь публики несколькими фишками:

- Встроенная асинхронность (о случаях ее использования читайте в [1.6 База данных: запросы](#))
- Встроенная документация к API в формате OpenAPI (ранее Swagger)
- Встроенная в фреймворк валидация данных через библиотеку Pydantic
- Простота освоения
- Скорость написания приложений
- Подробная документация
- Удобная работа с вебсокетами (в отличие, например, от Django Channels)

При возникновении у вас любого вопроса по фреймворку в первую очередь рекомендую обращаться к документации к фреймворку: <https://fastapi.tiangolo.com/>

Документация к API

Документация формата OpenAPI автоматически генерируется при использовании фреймворка FastAPI и выглядит так для нашего проекта. Мы вместе с вами напишем эти и многие другие эндпоинты

Бронирование Отелей ¹ OAS3

/api/v1/openapi.json

Servers

Auth ^

- POST** `/auth/register` Register User ▾
- POST** `/auth/login` Login User ▾
- POST** `/auth/logout` Logout User ▾

Пользователи ^

- GET** `/users/me` Read Users Me ▾

Отели ^

- GET** `/hotels/{location}` Get Hotels By Location And Time ▾
- GET** `/hotels/id/{hotel_id}` Get Hotel By Id ▾
- GET** `/hotels/{hotel_id}/rooms` Get Rooms By Time ▾

Она позволяет другим разработчикам (обычно фронтендерам и разработчикам мобильных приложений) работать с вашим API. В документации можно увидеть

- Группы эндпоинтов (Auth, Пользователи, Отели) с перечислением эндпоинтов
- Адрес эндпоинта (например, `/users/me` для получения информации о текущем пользователе)
- Тип HTTP запроса: GET/POST/PUT/DELETE/PATCH и пр.

- Требуемые данные для ввода
- Структура данных, которую отдает наш API
- Виды возможных ошибок для данного эндпоинта