

Фаза 3 • Неделя 1 • Понедельник

Fast API

Описание способов взаимодействия одной компьютерной программы с другими

Иначе говоря – как правильно обратиться к сервису, чтобы он вернул нам ожидаемый ответ

Уже было:

- `bot.send_message( )`
- `huggingface API`

1. **Отделение клиента от сервера (Client-Server).** Клиент — это пользовательский интерфейс сайта или приложения, например, поисковая строка видеохостинга. В REST API код запросов остается на стороне клиента, а код для доступа к данным — на стороне сервера.

1. Отделение клиента от сервера (Client-Server).
2. **Отсутствие записи состояния клиента (Stateless).** Сервер не должен хранить информацию о состоянии (проведенных операций) клиента. Каждый запрос от клиента должен содержать только ту информацию, которая нужна для получения данных от сервера.

например, для классификации картинки: токен доступа, картинка

1. Отделение клиента от сервера (Client-Server).
2. Отсутствие записи состояния клиента (Stateless).
3. Кэшируемость (Cacheable) (пока неважно)
4. Предоставление кода по запросу (Code on Demand) (пока неважно)

# Restful • REST API • Принципы

1. Отделение клиента от сервера (Client-Server).
2. Отсутствие записи состояния клиента (Stateless).
3. Кэшируемость (Cacheable) (пока неважно)
4. Предоставление кода по запросу (Code on Demand) (пока неважно)
5. **Начало от нуля (Starting with the Null Style)**. Клиент знает только одну точку входа на сервер. Дальнейшие возможности по взаимодействию обеспечиваются сервером.

например: адрес и порт сервера, нужный маршрут ( `bot.send_message()` )

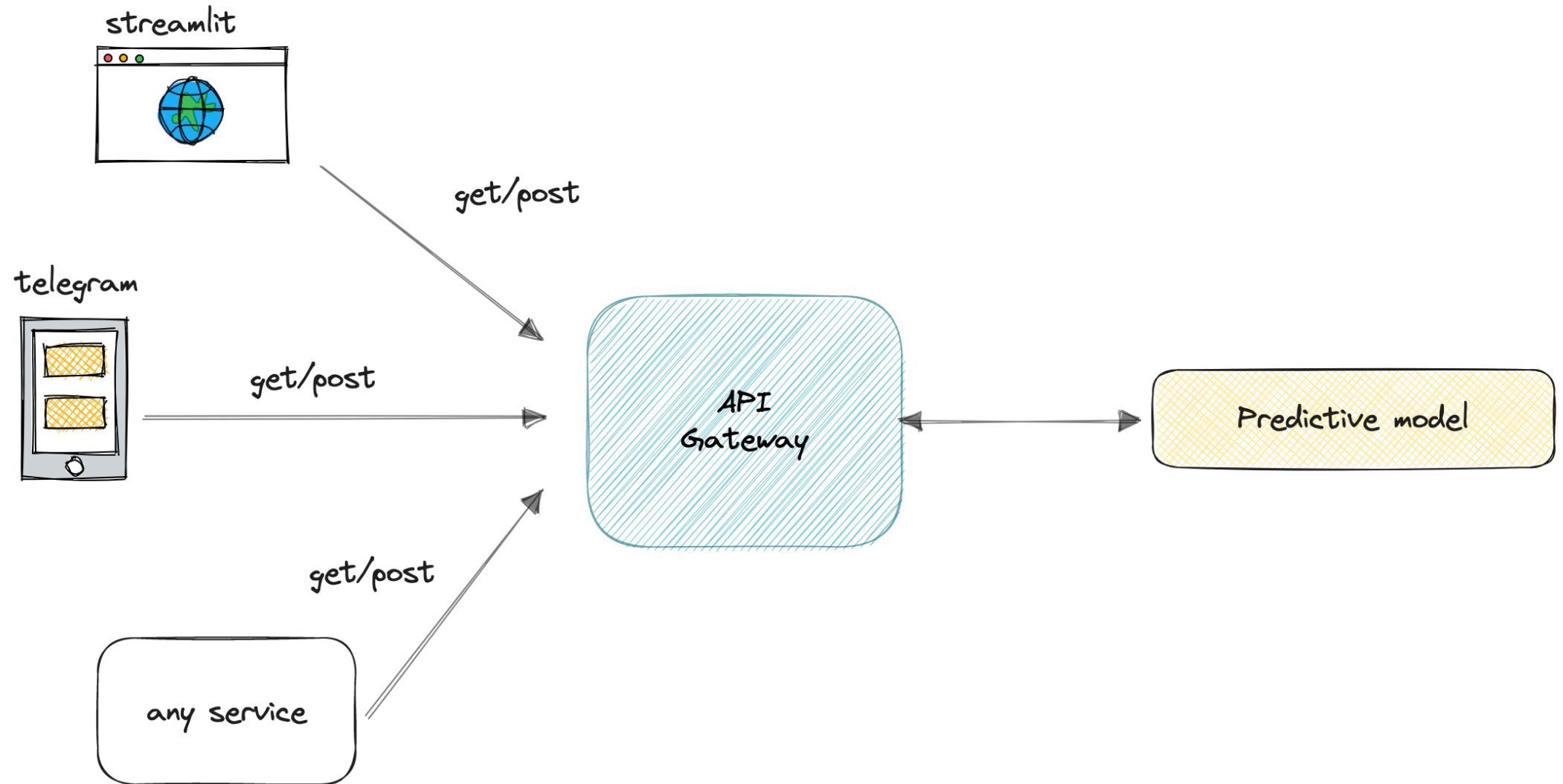
# HTTP-протокол • Запросы

В REST API есть 4 метода HTTP, которые используют для действий с объектами на серверах:

- **GET** (получение информации о данных или списка объектов)
- **DELETE** (удаление данных)
- **POST** (добавление или замена данных)
- **PUT** (регулярное обновление данных)

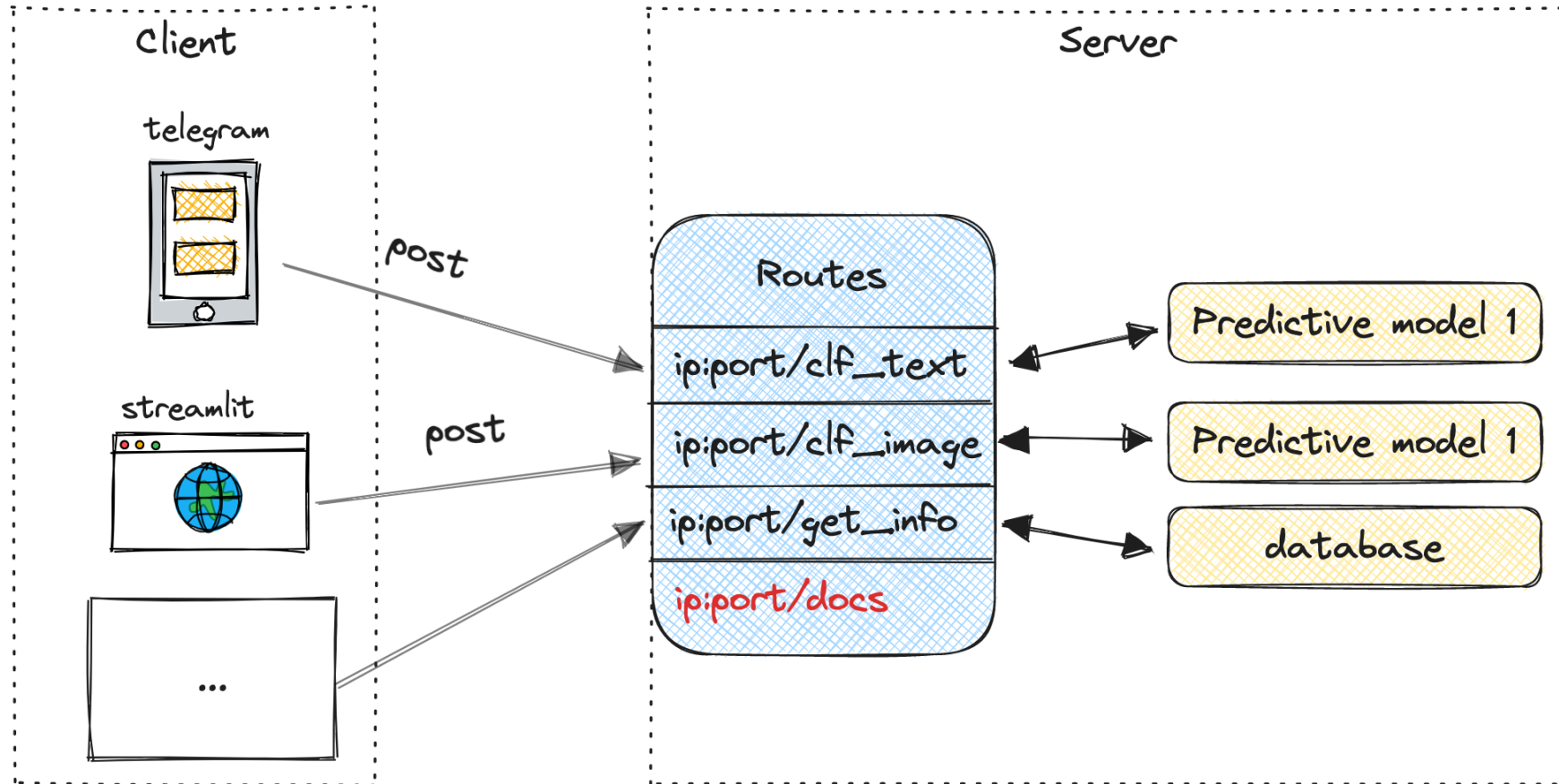
Кто занимался парсингом, должен помнить `requests.get()` – это и есть GET-запрос.

# Схема





# Упрощенная архитектура



- Разработка API – отдельная работа, в которую входит проектирование системы и нагрузки, кэширования и пр. Но иногда и это надо уметь делать.
- Популярный python-фреймворк – FastAPI:
  - Быстрый
  - Асинхронный
  - Понятный
- Можно почитать, например, это: <https://github.com/twirl/The-API-Book/tree/gh-pages>