

Тестовое задание на позицию Go-middle

Подзадание 1 - Rate Limmiter

Создать Rate Limmiter, HTTP-сервис, ограничивающий количество запросов из одной подсети IPv4. При отсутствии ограничений выдавать одинаковый, статический контент.

Общие требования:

Рекомендуем не тратить больше 4-5 часов на выполнение, рабочий сервис нам не нужен, хотим понять ход мыслей. Код следует снабдить подробными комментариями и пояснениями.

Функциональные требования:

- код доступен на GitHub
- ответ соответствует спецификации RFC 6585 (429 Too Many Requests etc.)
- IP извлекается из заголовка X-Forwarded-For, подсеть /24 (255.255.255.0)
- лимит 100 запросов в минуту
- cooldown после ограничений 1 минута

Пример: после 32 запросов с 123.45.67.89 и 68 запросов с 123.45.67.1 возвращем 429 ошибку на любой запрос с подсети 123.45.67.0/24 в течение последующей минуты.

Усложнения:

- контейнеризация, docker-compose up
- покрыть тестами
- размер префикса подсети, лимит и время cooldown'a задаются через переменные окружения и вычитываются при старте
- handler для сброса лимита по префиксу

Подзадание 2 - Clickhouse

Общие требования:

Ответ в текстовом файле с понятным форматированием.

Есть таблица ads_data:

Колонка	Комментарий
date	день, в который произошло событие
time	время события
event	тип события (показ или клик)
platform	платформа, на которой произошло событие
ad_id	идентификатор объявления
client_union_id	идентификатор клиента
campaign_union_id	идентификатор рекламной кампании
ad_cost_type	тип объявления с оплатой за клики (CPC) или за показы (CPM)
ad_cost	стоимость объявления в рублях, для CPC объявлений - это цена за клик, для CPM - цена за 1000 показов
has_video	есть ли у рекламного объявления видео
target_audience_count	размер аудитории, на которую таргетируется объявление

Написать следующие запросы:

1. Статистика по дням. Посчитать:

- число событий по дням
- число показов
- число кликов

- число уникальных объявлений
- уникальных кампаний.

2. Найти объявления по которым показ произошел после клика