**e-Comet.io: Тестовое задание для Backend Developer (Python+SQL) v2**

1. Допишите часть кода, чтобы зависимость *get\_pg\_connection* действительно возвращала подключение к PostgreSQL в обработчике *get\_db\_version.*
   1. Запрещено использовать глобальные переменные и объекты, а также любые механизмы, которые сохраняют данные вне локального контекста функций или методов
   2. Необходимо использовать пул подключений, который предоставляет библиотека [*asyncpg*](https://github.com/MagicStack/asyncpg)
   3. Запрещено использовать устаревшие (с пометкой deprecated) фичи FastAPI
2. Определите метод *get\_repositories* в классе *GithubReposScrapper*, который возвращает *Data*-класс *Repository*, где *authors\_commits\_num\_today* - список авторов с количеством коммитов за последний день.
   1. Отправлять запросы для получения коммитов репозитория АСИНХРОННО
   2. Ограничить максимальное количество одновременных запросов (*MCR*)
   3. Ограничить количество запросов в секунду (*RPS*)
3. На базе кода из 2-го задания реализуйте сохранение данных в *ClickHouse* о репозитории, его позиции в топе и количество коммитов авторов (схемы таблиц уже определены в приложении к заданию).
   1. Необходимо использовать библиотеку [*aiochclient*](https://github.com/maximdanilchenko/aiochclient)
   2. Эффективно использовать оперативную память с помощью вставки данных батчами
4. Напишите *SQL*-запрос в *ClickHouse*, который достаёт из заданной таблицы по определенной рекламной кампании просмотры по поисковым запросам по часам за сегодня.  
   Формат ответа должен быть следующим:  
   *phrase views\_by\_hour*  
   платье [(15, 4), (14, 6), (13, 4), (12, 1)]

Другие требования:

* Решение должно быть выложено на *GitHub* - одно репо, разбитое на фолдеры аналогично задачам
* Не объединять задачи в монолитное приложение, каждая задача (кроме 3) выполняется независимо от других
* Python 3.11+, PEP8
* Код должен быть безопасным, с обработчиками ошибок
* Значительное внимание уделить архитектуре и использовать современные решения
* Не использовать хардкод, вместо этого применять конфигурационные файлы, переменные окружения или другие механизмы настройки
* В целом это должен быть production-ready код, в том виде, в котором вы бы его рекомендовали к релизу