

# WB Tech: level # 0 (Golang)

## Тестовое задание

Необходимо разработать демонстрационный сервис с простейшим интерфейсом, отображающий данные о заказе. [Модель данных в формате JSON](#) прилагается к заданию.

Что нужно сделать:

1. Развернуть локально PostgreSQL
  - 1.1. Создать свою БД
  - 1.2. Настроить своего пользователя
  - 1.3. Создать таблицы для хранения полученных данных
2. Разработать сервис
  - 2.1. Реализовать подключение и подписку на канал в nats-streaming
  - 2.2. Полученные данные записывать в БД
  - 2.3. Реализовать кэширование полученных данных в сервисе (сохранять in memory)
  - 2.4. В случае падения сервиса необходимо восстанавливать кэш из БД
  - 2.5. Запустить http-сервер и выдавать данные по id из кэша
3. Разработать простейший интерфейс отображения полученных данных по id заказа

## Советы

1. Данные статичны, исходя из этого подумайте насчет модели хранения в кэше и в PostgreSQL. Модель в файле model.json
2. Подумайте как избежать проблем, связанных с тем, что в канал могут закинуть что-угодно
3. Чтобы проверить работает ли подписка онлайн, сделайте себе отдельный скрипт, для публикации данных в канал
4. Подумайте как не терять данные в случае ошибок или проблем с сервисом
5. Nats-streaming разверните локально (не путать с Nats)

## Бонус-задание

1. Покройте сервис автотестами — будет плюсик вам в карму.
2. Устройте вашему сервису стресс-тест: выясните на что он способен.

Воспользуйтесь утилитами WRK и Vegeta, попробуйте оптимизировать код.

# Результат

По готовности сервиса снимите короткое видео работы интерфейса и вместе со ссылкой на репозиторий пришлите в телеграм: [@avkonov](#)