**Обязательные требования**

• Использование go fmt и goimports

• Следование Effective Go

• Go актуальной версии

• Использование JSON-API. Каждая операция должна быть RPC-like, то есть выполнять определенное законченное действие.

• PostgreSQL или MySQL в качестве хранилища данных

• Наличие команды make up в Makefile, которая: поднимает без ошибок приложение с помощью Docker контейнеров, и готовую инфраструктуру для работы приложения (база данных, миграции, данные для тестирования работы приложения)

• Описание API методов с работающим запросом и ответом в одном из следующих форматов: .http файлы (IDEA) с ответами, curl команды с ответами в README.md, коллекция Postman, построенная на основе swagger / openapi коллекции.

**Критерии оценки**

• Работоспособность API

• API выполняет заявленные функции

• API предусматривает граничные кейсы

• Нахождение и решение потенциальных проблем

• Организация и читаемость кода

• Обработка ошибок

Будет плюсом

• Покрытие кода unit или функциональными тестами

• Аргументация выбора пакетов в go.mod, приложить отдельным файлом packages.md

Результат

• Проект должен быть выложен в публичный репозиторий Github/Gitlab

• В проекте должен присутствовать README и содержать в себе:

• Инструкцию по запуску сервиса

• Инструкцию по запуску тестов при их наличии

**Задание**

**#1. API для работы с товарами на складе**

Необходимо спроектировать и реализовать API методы для работы с товарами на одном складе.   
Учесть, что вызов API может быть одновременно из разных систем и они могут работать с одинаковыми товарами.  
Методы API можно расширять доп. параметрами на своё усмотрение

• Спроектировать и реализовать БД для хранения следующих сущностей

• Склад

• название

• признак доступности

• Товар

• название

• размер

• уникальный код

• количество

• Реализовать методы API:

• резервирование товара на складе для доставки

• на вход принимает:

• массив уникальных кодов товара

• освобождение резерва товаров

• на вход принимает

• массив уникальных кодов товара

• получение кол-ва оставшихся товаров на складе

• на вход принимает:

• идентификатор склада

• Будет плюсом

• Реализация логики работы с товарами, которые одновременно могут находиться на нескольких складах