**Тестовое задание web-программист Python (Middle)**

**#API: Сервис поиска ближайших машин для перевозки грузов.**

🔥 Необходимо разработать REST API сервиc для поиска ближайших машин к грузам.

◼Стек и требования:

* **Python** (Django Rest Framework / FastAPI) на выбор.
* **DB** - Стандартный PostgreSQL.
* Приложение должно запускаться в docker-compose без дополнительных доработок.
* Порт - 8000.
* БД по умолчанию должна быть заполнена 20 машинами.
* Груз обязательно должен содержать следующие характеристики:
  + локация pick-up;
  + локация delivery;
  + вес (1-1000);
  + описание.
* Машина обязательно должна в себя включать следующие характеристики:
  + уникальный номер (цифра от 1000 до 9999 + случайная заглавная буква английского алфавита в конце, пример: "1234A", "2534B", "9999Z")
  + текущая локация;
  + грузоподъемность (1-1000).
* Локация должна содержать в себе следующие характеристики:
  + город;
  + штат;
  + почтовый индекс (zip);
  + широта;
  + долгота.

*Список уникальных локаций представлен в прикрепленном csv файле "uszips.csv".*

[uszips.csv](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/73ce520f-5205-47d4-8169-2266c628f6a7/uszips.csv)

*Необходимо осуществить выгрузку списка в базу данных Postgres при запуске приложения.*

* При создании машин по умолчанию локация каждой машины заполняется случайным образом;
* Расчет и отображение расстояния осуществляется в милях;
* Расчет расстояния должен осуществляться с помощью библиотеки geopy. help(geopy.distance). Маршруты не учитывать, использовать расстояние от точки до точки.

⭐ Задание разделено на 2 уровня сложности. Дедлайн по времени выполнения зависит от того, сколько уровней вы планируете выполнить. **1 уровень** - 3 рабочих дня. **2 уровень** - 4 рабочих дня.

**◼Уровень 1**

Сервис должен поддерживать следующие базовые функции:

* Создание нового груза (характеристики локаций pick-up, delivery определяются по введенному zip-коду);
* Получение списка грузов (локации pick-up, delivery, количество ближайших машин до груза ( =< 450 миль));
* Получение информации о конкретном грузе по ID (локации pick-up, delivery, вес, описание, список номеров ВСЕХ машин с расстоянием до выбранного груза);
* Редактирование машины по ID (локация (определяется по введенному zip-коду));
* Редактирование груза по ID (вес, описание);
* Удаление груза по ID.

**◼Уровень 2**

Все что в уровне 1 + дополнительные функции:

* Фильтр списка грузов (вес, мили ближайших машин до грузов);
* Автоматическое обновление локаций всех машин раз в 3 минуты (локация меняется на другую случайную).

**◼Критерии оценки:**

* Адекватность архитектуры приложения;
* Оптимизация работы приложения.

💡 Результат тестового задания необходимо отправить HR-менеджеру Екатерине ответным сообщением. Ссылка на репозиторий в <https://github.com/> .

* ❓**FAQ**
  + **Я делал(-а) похожее задание, можете его проверить?**

Мы проверяем лишь наше тестовое задание.

* + **Я не успеваю за 72 часа, что делать?**

Напишите в чат, что не успеваете.

* + **Оплачивается ли тестовое задание?**

Оплата тестового задания обсуждается с кандидатом, успешно прошедшим все этапы отбора. Без выполнения тестового задания мы не сможем оценить Ваши знания и навыки по отношению к требованиям данной вакансии.

* + **Сколько еще этапов отбора?**

Если тестовое задание будет выполнено хорошо, Вас ждет техническое собеседование, а далее предложение о работе 😉

* + **Есть вопросы по тестовому заданию, куда их направить?**

К сожалению, в чате Вам не смогут ответить по поводу каких-либо особенностей выполняемого задания, поэтому действуйте в рамках ТЗ и можете экспериментировать

С уважением, компания WelbeX.