

# ТЕСТОВОЕ ЗАДАНИЕ на позицию Junior Python разработчик

## Задача

Реализовать веб-сервис на базе django, предоставляющий REST-api и способный:

1. Принимать из POST-запроса .csv файлы для дальнейшей обработки;
2. Обрабатывать типовые deals.csv файлы, содержащие истории сделок;
3. Сохранять извлеченные из файла данные в БД проекта;
4. Возвращать обработанные данные в ответе на GET-запрос.

## Требования

1. Данные хранятся в реляционной БД, взаимодействие с ней осуществляется посредством django ORM.
2. Ранее загруженные версии файла deals.csv не должны влиять на результат обработки новых.
3. Эндпоинты соответствуют спецификации:

### Выдача обработанных данных

Метод: GET

В ответе содержится поле "response" со списком из 5 клиентов, потративших наибольшую сумму за весь период.

Каждый клиент описывается следующими полями:

- username - логин клиента;
- spent\_money - сумма потраченных средств за весь период;
- gems - список из названий камней, которые купили как минимум двое из списка "5 клиентов, потративших наибольшую сумму за весь период", и данный клиент является одним из этих покупателей.

### **Загрузка файла для обработки**

Метод: POST

Аргументы:

- deals: файл, содержащий историю сделок.

Ответ:

- Status: OK - файл был обработан без ошибок;
- Status: Error, Desc: <Описание ошибки> - в процессе обработки файла произошла ошибка.

4. Приложение должно быть контейнеризировано при помощи docker;
5. Проект не использует глобальных зависимостей за исключением: python, docker, docker-compose;
6. Readme проекта описывает весь процесс установки, запуска и работы с сервисом;
7. Требования к фронтенду не предъявляются, интерфейс взаимодействия - RestFul API;
8. Проект запускается одной командой.

### **Будет плюсом**

1. Команда, используемая для запуска проекта - `docker-compose up`;
2. Кэширование данных, возвращаемых GET-эндпоинтом, с обеспечением достоверности ответов;
3. Сервис django работает на многопоточном WSGI-сервере;
4. API реализован на основе DRF.

## Файлы

**deals.csv** - содержит историю сделок по продаже камней. Описание полей deals.csv:

- customer - логин покупателя
- item - наименование товара
- total - сумма сделки
- quantity - количество товара, шт
- date - дата и время регистрации сделки

Рекомендуемое время выполнения тестового задания: 2-3 дня

## Полезные ссылки

- Документация django - <https://docs.djangoproject.com/en/3.0/>
- Документация DRF - <https://www.django-rest-framework.org/>
- Документация docker - <https://docs.docker.com/>
- Документация docker-compose - <https://docs.docker.com/compose/>
- Немного о качестве кода - <https://realpython.com/python-code-quality/>
- Python Debugger - <https://docs.python.org/3/library/pdb.html>