

Тестовое задание на позицию Python Developer

Общая информация

1. Тестовое задание разделено на две части. Часть №1 является задачей на логическое мышление и умения построения алгоритмов. В части №2 необходимо разработать небольшое WEB-приложение с использованием Django.

2. Вы должны отправить Ваше решение для тестового задания на нашу электронную почту amasales.service@gmail.com **в виде ссылки на github репозиторий**. Если Вы пришлете Ваше решение в любом другом виде (в виде ссылки на zip архив, прикрепите zip архив к письму, и др.) – Ваша кандидатура не будет нами рассмотрена!

3. Вам необходимо также приложить инструкцию по установке и запуску проекта (можно добавить в README.md на GitHub'e), ФИО.

4. Укажите в письме список выполненных пунктов для каждой части. Если Вы сделали не все пункты тестового задания – пожалуйста, укажите причину по которой вы их не выполнили (не хватило времени, не хватает опыта/знания, что-то ещё).

Технические требования для второй части ТЗ

- Django 3.1+, DRF 3+
- MySQL 5.6+ / PostgreSQL 12+
- Python 3.8+
- aiohttp 3.7+
- PyDantic 1.9+

При выполнении тестового задания Вы можете дополнительно использовать любые сторонние Python без всяких ограничений. Все 3rd party Python библиотеки должны быть добавлены в проект через pip если библиотека поддерживает такой способ установки.

Часть №1

Дан массив связей пользователей. Вам необходимо реализовать функцию, которая принимает на вход три аргумента: информация о связях, как кортеж (tuple) кортежей, первое имя (str), второе имя (str). Функция должна возвращать True, если связь между любыми двумя заданными пользователями существует, например, если у двух пользователей есть общие друзья или у их друзей есть общие друзья и т.д., иначе False.

```
1: def check_relation(net, first, second):
2:     pass
3:
4:
5: if __name__ == '__main__':
6:     net = (
7:         ("Ваня", "Лёша"), ("Лёша", "Катя"),
8:         ("Ваня", "Катя"), ("Вова", "Катя"),
9:         ("Лёша", "Лена"), ("Оля", "Петя"),
10:        ("Стёпа", "Оля"), ("Оля", "Настя"),
11:        ("Настя", "Дима"), ("Дима", "Маша")
12:    )
13:
14:    assert check_relation(net, "Петя", "Стёпа") is True
15:    assert check_relation(net, "Маша", "Петя") is True
16:    assert check_relation(net, "Ваня", "Дима") is False
17:    assert check_relation(net, "Лёша", "Настя") is False
18:    assert check_relation(net, "Стёпа", "Маша") is True
19:    assert check_relation(net, "Лена", "Маша") is False
20:    assert check_relation(net, "Вова", "Лена") is True
```

Часть №2

1. Изначально необходимо отследить трафик с www.wildberries.ru и найти HTTP запрос, который в JSON формате присылает данные о бренде и название артикула. Пример страницы товара:
<https://www.wildberries.ru/catalog/73512949/detail.aspx>



Мир Фигурного Катания / Аксессуар для коньков

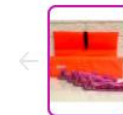
Название артикула



0 отзывов Артикул: 73512949



Цвет: оранжевый витамин



Все товары Мир Фигурного Катания >

Все аксессуары для коньков Мир Фигурного Катания >

Бесплатная доставка

21 день на возврат



Мир Фигурного Катания

Бренд

2. Далее необходимо реализовать API принимающее файл формата `xlsx` с артикулами (артикулы должны вводиться построчно в первой колонке) или один артикул (не в файле, а исключительно одно значение). В API должно быть два инпута: файл или одно значение, передаваться должно что-то одно.
3. API должно асинхронно взаимодействовать с найденным HTTP запросом в первом пункте и получать данные о карточке товара. Из полученных данных необходимо сделать PyDantic объект.
4. Успешным результатом работы API является возврат данных о бренде и названии артикула в JSON формате. Пример: информация об одном артикуле - `{"article": 123, "brand": "brand", "title": "Title"}`; артикулы из файла - `[{"article": 1, "brand": "Brand1", "title": "Title1"}, {"article": 2, "brand": "Brand2", "title": "Title2"}]`

Пожалуйста не забудьте, что Ваше тестовое задание должно быть предоставлено в виде ссылки на **github репозиторий**. Мы рассмотрим каждое задание, соответствующее заявленным выше требованиям. Рассмотрение занимает некоторое время, поэтому не стоит перезванивать нам на следующий день, чтобы узнать результат. Так же не стоит спешить и отправлять только часть выполненных заданий, если чувствуете, что можете сделать больше. Мы в поиске талантов, проявите себя! Мы желаем Вам удачи и с нетерпением ждем Ваше выполненное задание.