

Тестовое задание на позицию Python Developer (Technical Support)

В этом тестовом задании Вам нужно разработать онлайн каталог сотрудников для компании.

Общая информация :

1. Тестовое задание разделено на две части. Часть №1 обязательная и мы примем Вашу кандидатуру к рассмотрению только в том случае если часть №1 выполнена. Часть №2 опциональная, по количеству задач из этой части которые Вы смогли выполнить мы сможем оценить Ваш уровень опыта и навыков в области веб разработки. Попытайтесь выполнить все или как можно больше задач из части №2, это повысит ваши шансы по отношению к другим кандидатам.
2. Вы должны отправить Ваше решение для тестового задания на нашу почту techsupreview@rocketdata.io 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 ссылку или вложением подробное резюме, контактный номер телефона для связи по средствам telegram.
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:

- Python 3.7+
- Django 2.1+
- DRF 3.10+
- Celery 4.4 +
- PostgreSQL 10+
- Docker

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

Часть №1 (обязательная)

Задание:

- Создайте веб-приложение , с API интерфейсом и админ-панелью.
- Создайте базу данных используя миграции Django

Требования:

1. Информация о каждом сотруднике должна храниться в базе данных и содержать следующие данные:
 - ФИО;
 - Должность;
 - Дата приема на работу;
 - Размер заработной платы;
 - Информация о выплаченной зарплате;
2. У каждого сотрудника есть 1 начальник;
3. База данных должна содержать 5 уровней иерархий;
Под иерархией понимается древовидная структура на подобие структурной схемы предприятия. Например:
 - а) У генерального директора нет руководителей, но есть подчинённые. Его уровень иерархии будет равен 0;
 - б) У менеджера среднего звена уровень иерархии 2. Все его подчинённые менеджеры будут иметь уровень иерархии 3, а у его руководителя – уровень 1.Важно отметить, что уровень иерархии определяется не названием должности, а отношением к остальным сотрудникам.
4. На странице сотрудников в админ-панели должны выводиться:
 - ФИО.
 - Должность.
 - Ссылка на объект начальника.
 - Зарботная плата.
 - Всего выплачено.
5. На странице сотрудников в админ-панели должны быть фильтры:
 - Должность.
 - Уровень.
6. На странице сотрудников в админ-панели должен быть action который удаляет всю информацию о выплаченной заработной плате всех выбранных сотрудников.
7. Используя DRF, создать набор представлений:
 - Все данные о сотрудниках.
 - Все данные о сотрудниках одного уровня.
8. Сотрудники с определенной группой прав должны иметь доступ к API;

Часть №2 (опциональная)

Задание:

- Заполнить базу данных.
- Автоматизировать процессы в приложении при помощи Celery.
- Настроить отдельный доступ к API.
- Настройте автоматическое развертывание проекта.

Требования:

1. Заполните базу данных при помощи:
 - DB seeder для Django ORM
 - При помощи скрипта.
2. Напишите несколько celery задач, что бы упростить работу с приложением в админ-панели:
 - Задача должна запускаться автоматически каждые 2 часа и начислять сотруднику заработную плату.
 - Если action применяется более чем к 20 сотрудникам, стирание данных должно производиться асинхронно.
3. Пользователи могут получать доступ к API при помощи специального ключа.
 - Пользователь может получить информацию только о себе.
4. Проект должен разворачиваться при помощи файла docker-compose.