

Тестовое задание

Main technologies: python3 + django + django-rest-framework

VCS: Git

DB: SQLite

Deploy: docker compose

Необходимо построить REST API service.

Роли пользователей: regular user, admin user

Авторизация: django-rest-framework auth token

Формат коммуникации с API: JSON

Сервис будет позволять клиентам регистрировать и проводить управление компанией, её офисами, сотрудниками, автопарком этой компании.

Основные сущности БД:

Company

User

Vehicle

Office

Requirements:

1. Registration of user and company

Я могу зарегистрировать свой аккаунт и свою компанию в сервисе. Я получаю роль админа в этой компании.

Пользователь должен передать след информацию: first_name, last_name, email, password, repeat_password, company_name.

2. Login

Я могу залогиниться (т.е получить auth token), передав свои email и password

3. Workers register/list

Я могу, как администратор своей компании, зарегистрировать в ней сотрудника - такой же пользователь как и я, но без “admin” прав и посмотреть список всех моих сотрудников.

Могу отфильтровать сотрудников по first_name, last_name, email.

4. Worker view/update/delete

Я могу, как администратор своей компании, посмотреть данные своего работника (first_name, last_name, email); изменить их (включая пароль), но НЕ email; удалить сотрудника.

5. Company view

Я могу, как сотрудник или администратор своей компании, посмотреть её name, address (CharField)

6. Company edit

Я могу, как администратор своей компании, поменять её name, address

7. MyProfile view/edit

Я могу, как сотрудник или администратор, посмотреть свой профиль (first_name, last_name, email) и поменять first_name, last_name и password

8. Create office/view list of offices

Я могу, как администратор своей компании, создать для неё офис (name, address, country, city, region). Как сотрудник или администратор, посмотреть список офисов моей компании. Могу отфильтровать офисы по стране их размещения и/или городу.

9. Office retrieve/update/delete

Я могу, как администратор своей компании, посмотреть информацию по конкретному офису, отредактировать данные офиса(name, address, country, city, region), удалить этот офис.

10. Assign worker to an office

Как администратор компании, я могу назначить сотрудника в офис. Сотрудник НЕ может работать более чем в одном офисе. Сотрудник МОЖЕТ быть НЕ назначен ни в один офис.

11. View my office

Как сотрудник, я могу посмотреть детали офиса в который меня назначили.

12. Vehicle list/create

Как администратор компании, я могу создать запись о транспортном средстве (licence_plate, name, model, year_of_manufacture) , что есть на балансе моей компании. Опционально, могу указать к какому офису относится это транспортное средство и какой сотрудник есть его

водителем. (Важно, что сотрудник ДОЛЖЕН быть в том же офисе что и транспортное средство).

Могу посмотреть список транспорта в моей компании, отфильтровать по офису и\или водителю.

13. Vehicle view/update/delete

Как администратор компании, я могу посмотреть детали транспортного средства, отредактировать их, удалить это транспортное средство.

14. My Vehicles

Как сотрудник компании, я могу посмотреть список транспортных средств, водителем которых я являюсь.

Дополнительные требования:

1. Соблюдения правил построения ресурсов REST API
2. Соблюдение best practices: DRY, KISS, python-zen
3. Верное разбиение проекта по модулям, пакетам
4. Учитывать, что в системе может быть несколько администраторов и их компаний и они не должны видеть НИКАКОЙ информации о друг-друге.
5. По завершению работы подготовить демо, на котором продемонстрировать работу системы.
(Как вариант - с применением Postman)