

Тестовое задание Backend Python (ProninTeam)

Задача Требования Полезные ссылки

Задача

Реализовать веб-сервис на базе Django, предоставляющий *CRUD REST API* для групповых денежных сборов. Приложение должно содержать **как минимум** три сущности:

- User (модель Пользователя системы, можно использовать стандартного)
- Payment (модель описывающая Платёж для сбора)
- Collect (модель непосредственно Группового денежного сбора).

Групповой сбор должно должен содержать как минимум:

- автора сбора
- название
- повод (один из нескольких на выбор, например, "день рождение", "свадьба" и т.д.)
- описание
- сумму, которую запланировали собрать (можно сделать "бесконечный" сбор, вез верхней границы)
- сумму, которую собрали на текущий момент
- сколько человек уже сделало пожертвования
- обложку сбора (картинка)

- дата и время завершения сбора
- лента сбора (список пожертвований от других пользователей: сумма, дата и время, ФИО пользователя)

При создании *Группового сбора* или *Платежа по сбору* на почту автора/ донатера должно прийти письмо с информацией об успешном создании сбора (отправке платежа).

Все эндпоинты соответствуют REST и покрыты документацией Swagger.

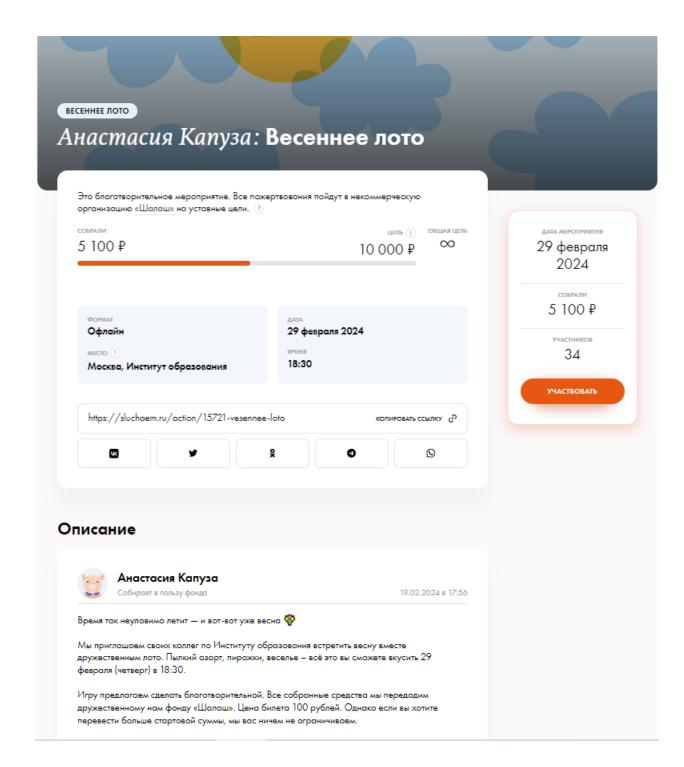
Требования

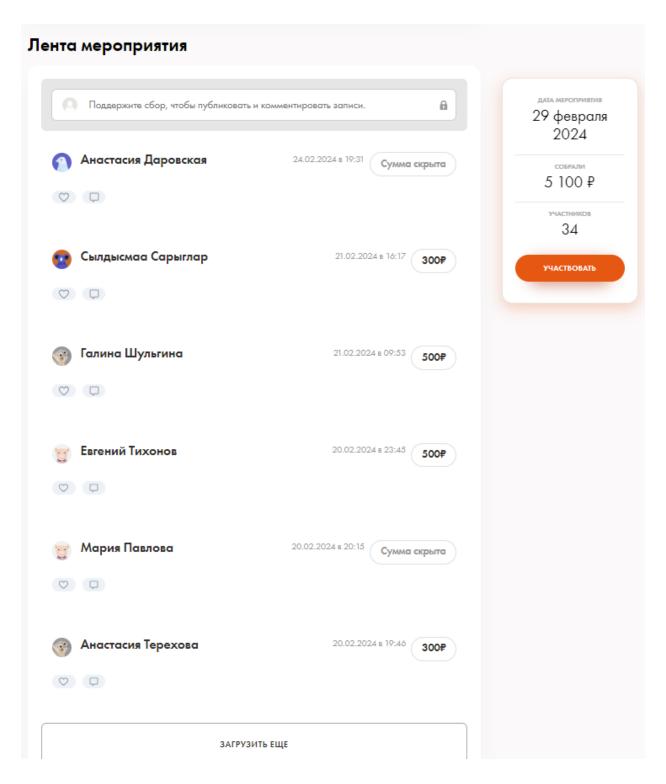
- 1. Данные хранятся в реляционной БД, взаимодействие с ней осуществляется посредством Django ORM.
- 2. API реализовано на базе Django REST Framework
- 3. Реализовано кэширование данных, возвращаемых GET-эндпоинтом, с обеспечением достоверности ответов
- 4. Проект должен быть докеризирован и запускаться через docker compose up.
- 5. Должна присутствовать <u>Management command</u> для наполнения БД моковыми данными (несколько тысяч).

Можно использовать любые дополнительные библиотеки.

Рекомендуемое время выполнения тестового задания: 3 дня

Референс: https://sluchaem.ru/action/15721-vesennee-loto





Полезные ссылки

- Документация DRF https://www.django-rest-framework.org/
- Документация docker https://docs.docker.com/

- Документация docker-compose https://docs.docker.com/compose/
- Celery https://docs.celeryq.dev/en/stable/
- Redis https://pypi.org/project/django-redis/