Дополнительные советы ментора

Реализация модели данных

В пакете `item` создайте класс `Item` и добавьте в него следующие поля:

- `id` — уникальный идентификатор вещи;
- `name` — краткое название;
- `description` — развёрнутое описание;
- `available` — статус о том, доступна или нет вещь для аренды;
- `owner` — владелец вещи;
- `request` — если вещь была создана по запросу другого пользователя, то в этом поле будет храниться ссылка на соответствующий запрос.

Класс `User` в пакете `user` будет совсем простым. Всего три поля:

- `id` — уникальный идентификатор пользователя;
- `name` — имя или логин пользователя;
- `email` — адрес электронной почты (учтите, что два пользователя не могут иметь одинаковый адрес электронной почты).

Класс `Booking` в пакете `booking` будет содержать следующие поля:

- `id` — уникальный идентификатор бронирования;
- `start` — дата и время начала бронирования;
- `end` — дата и время конца бронирования;
- `item` — вещь, которую пользователь бронирует;
- `booker` — пользователь, который осуществляет бронирование;
- `status` — статус бронирования. Может принимать одно из следующих значений: `WAITING` — новое бронирование, ожидает одобрения, `APPROVED` —

бронирование подтверждено владельцем, `REJECTED` — бронирование отклонено владельцем, `CANCELED`— бронирование отменено создателем.

И в пакете `request` будет `ItemRequest` — класс, отвечающий за запрос вещи. Его атрибуты:

- `id` — уникальный идентификатор запроса;
- `description` — текст запроса, содержащий описание требуемой вещи;
- `requestor` — пользователь, создавший запрос;
- `created` — дата и время создания запроса.

Добавьте необходимые аннотации Lombok, чтобы сгенерировать геттеры и сеттеры для полей.

Создание DTO-объектов и мапперов

В пакетах `item` и `user` создайте дополнительный DTO-класс (например, `ItemDto`). Для начала можно оставить в DTO те же поля, что и в изначальной модели. Когда вы поймёте, какую именно структуру данных контроллеры должны возвращать пользователю или получать от него, можно будет добавить новые атрибуты.

Для сущностей `Item` и `User` создайте Mapper-класс и метод преобразования объекта модели в DTO-объект. Например, для сущности `Item` вам нужно создать класс `ItemMapper` и добавить в него метод `toItemDto()`, который может выглядеть примерно так.

Листинг

```
public static ItemDto toItemDto(Item item) {
    return new ItemDto(
            item.getName(),
            item.getDescription(),
            item.isAvailable(),
            item.getRequest() != null ? item.getRequest().getId() : null
    );
}
```

По аналогии создайте маппер для `User`. В дальнейшем можно будет добавлять в них методы, которые нужны для обратного преобразования типов, — например, `toItem().`