

Чек-лист по самопроверке №22

Перед отправкой работы на ревью убедитесь, что она соответствует всем критериям, которые описаны в этом документе.

Работа отклоняется от ревью

Ревьюеры не станут проверять работу, если она соответствует хотя бы одному критерию из этого блока:

- Ошибки при выполнении команды запуска проекта.
- Отсутствует корректно заполненный файл `.gitignore`.
- Реализована не вся обязательная функциональность.
- Работа содержит вопросы или просьбы о помощи к ревьюеру.
- На повторных итерациях не исправлены критические замечания.

Работа принимается

В этом блоке собраны требования к работе, по которым вы можете выявить и самостоятельно исправить частые ошибки.

Файловая структура проекта:

- В проекте есть файл `package.json`:
 - В `package.json` корректно заполнены поля `name` и `author`;
 - `package.json` содержит все зависимости, используемые в проекте;
 - В `package.json` нет лишних зависимостей;
 - в разделе `scripts` файла `package.json` описаны команды запуска проекта, созданные при развертывании заготовки через Nest CLI.
- В проекте есть файл `.gitignore`, как минимум в этот файл добавлена папка `node_modules`.
- У проекта ясная структура, файлы, относящиеся к одной сущности, расположены в соответствующей папке, например, папки `offers`, `users`, `wishes`, `wishlists`. Каждая папка содержит:
 - файлы контроллера, сервиса, модуля;
 - папку `dto` с файлами, описывающими объекты передачи данных;
 - папку `entities` с файлом, описывающим модель данных.
- Нет лишних файлов, которые не относятся к проекту.

Работа приложения:

- Реализованы все запросы из описания API сервиса.
- Форматы запросов и ожидаемый ответ сервиса соответствуют документации на API.
- При разных запросах сервер не падает, и в консоли нет ошибок.
- Настроено подключение к базе данных с помощью TypeORM. В настройках подключения установлен флаг `synchronize: true` либо описаны миграции.
- В контролерах описаны необходимые по заданию эндпоинты запросов. Контролеры не взаимодействуют с базой данных напрямую.
- Взаимодействие с базой данных описано в соответствующих сервисах.
- Описаны модели данных для пользователей (`users`), подарков (`wishes`), списков желаний (`wishlists`) и списка желающих скинуться (`offers`).

- Для каждого из методов контроллера описан DTO, а также указаны валидаторы. Валидация реализована через `ValidationPipe` и `class-validator`.
- Поля моделей валидируются с помощью декораторов из пакета `class-validator`.
- Правильно настроены связи моделей, то есть применяется верный тип связи: `OneToOne`, `OneToMany`, `ManyToOne` или другой.
- Регистрация и авторизация
 - Реализована регистрация и авторизация с применением пакетов `@nestjs/passport`, `@nestjs/jwt`, `passport`, `passport-local` и `bcrypt`.
 - Пароли хранятся в хешированом виде.
 - API не возвращает хеш пароля пользователя и его почту (за исключением метода получения профиля пользователя).
 - Для полей email и username заданы требования уникальности, ошибки корректно обрабатываются в контроллере регистрации и обновления профиля пользователя.
 - Для JWT-токена задано время жизни.
 - Все запросы, кроме запроса 40 последних подарков, запроса 10 популярных подарков, запроса регистрации и авторизации, защищены гардом авторизации через декоратор метода `UseGuards`.
- Реализованы все ограничения в работе сервиса, описанные в задании:
 - пользователь может редактировать только свой профиль;
 - редактировать и удалять можно только свои подборки подарков;
 - нельзя изменять или удалять чужие подарки;
 - нельзя изменять стоимость подарка, если уже есть желающие скинуться;
 - сумма собранных средств недоступна для изменения и формируется из заявок других пользователей;
 - пользователю нельзя вносить деньги на собственные подарки, а также удалять или редактировать заявки;
 - нельзя скинуться на подарки, на которые уже собраны деньги. Сумма собранных средств не может превышать стоимость подарка.
- Фронтенд проекта корректно работает с разработанным бэкендом.

Типизация проекта реализована верно:

- Установлены все необходимые пакеты для `TypeScript`.
- Корректно работает линтинг `ts`-файлов.
- Конфигурация `tsconfig.json` настроена верно, нет настроек, ослабляющих контроль типов.
- `any` не используются без необходимости.
- Инлайновое отключение линтера не используется без необходимости.
- Описания одинаковых типов не дублируются.

Линтинг:

- При запуске команды `npm run lint` выполняется проверка проекта, по её результатам не должно быть ошибок линтинга.