

WEB. Разработка с Nest JS.

Урок 23. Практика. API Gateway.

Техническое задание: Замена NGINX на самописный API Gateway.

Цель: Заменить существующий сервис NGINX на самописный API Gateway, реализованный на базе Node.js с использованием библиотеки Axios для проксирования запросов и сборки данных.

Основная задача нового API Gateway — роутинг запросов к микросервисам, а также агрегация данных для отзывов.

Описание функциональности.

Общий роутинг запросов.

API Gateway должен перенаправлять запросы от клиентов к соответствующим микросервисам на основе маршрута.

Например:

- /users — запросы к сервису пользователей.
- /products — запросы к сервису товаров.
- /cart — запросы к сервису корзины.
- /chat — запросы к сервису чата.
- /reviews — запросы к сервису отзывов.

В большинстве случаев API Gateway должен просто передавать запросы напрямую к соответствующему микросервису без изменения или обработки.

Основная задача - обработка отзывов.

Для запросов к /reviews API Gateway должен собирать данные о пользователе и товаре, чтобы возвращать расширенную информацию.

Сервис отзывов будет возвращать отзыв в следующем формате:

```
{
  "user_id": "string",
  "good_id": "string",
  "report": {
    "title": "string",
    "report_info": "string",
    "rating": 5
  }
}
```

API Gateway должен:

Сделать запрос к сервису пользователей по `user_id` и получить информацию о пользователе.

Сделать запрос к сервису товаров по `good_id` и получить информацию о товаре.

Собрать итоговый объект с полными данными и вернуть его клиенту в формате:

```
{
  "user": {
    "id": "string",
    "name": "string",
    "email": "string",
    ...
  },
  "good": {
    "id": "string",
    "name": "string",
    "price": "number",
    ...
  },
  "report": {
    "title": "string",
    "report_info": "string",
    "rating": 5
  }
}
```

Проксирование запросов.

API Gateway должен проксировать запросы на соответствующие микросервисы:

- Любые запросы на `/users/*` — направлять на сервис пользователей.
- Запросы на `/products/*` — направлять на сервис товаров.
- Запросы на `/cart/*` — направлять на сервис корзины.
- Запросы на `/chat/*` — направлять на сервис чата.
- Запросы на `/reviews/*` — обрабатывать, как указано выше.

Подключение к микросервисам.

API Gateway должен использовать `Axios` для отправки запросов к микросервисам.

Каждый сервис доступен по определённому порту (например, сервис пользователей на порту 3001, сервис товаров на порту 3002 и т.д.).

Конфигурация для каждого сервиса (URL, порты) должна быть вынесена в переменные окружения или в отдельный конфигурационный файл.

Логика обработки отзывов.

Получить отзыв от сервиса отзывов по маршруту /reviews.

Извлечь user_id и good_id из ответа.

Сделать два дополнительных запроса к сервисам пользователей и товаров для получения полной информации о пользователе и товаре.

Собрать и вернуть объединённый объект, как описано выше.

Создание базовой структуры API Gateway.

Инициализировать новый проект на Node.js.

Установить необходимые зависимости: axios например.

Реализация базового проксирования запросов:

Настроить маршруты для пользователей, товаров, корзины и чата, чтобы просто перенаправлять запросы на соответствующие микросервисы.

Реализация агрегации данных для отзывов.

Реализовать логику получения данных от сервиса отзывов.

Настроить запросы к сервисам пользователей и товаров для получения данных по user_id и good_id.

Собрать данные и вернуть клиенту объединённый ответ.

Тестирование.

Провести тестирование маршрутов и убедиться, что API Gateway корректно взаимодействует с микросервисами.

Тестировать агрегацию данных для отзывов.