

Репозиторий - <https://github.com/artstesh/otus-highload/tree/mono>

Ветка - mono

Инструкция по запуску и тестированию

Предварительные требования

- Docker и Docker Compose
- Postman для тестирования API

Запуск системы

Перейдите в папку \deployment и запустите файл start.bat.

Тестирование функциональности

Коллекция Postman доступна по адресу <https://www.postman.com/orange-satellite-666437/otus-highload/collection/8uv29vb/otus-highload>

Комментарий:

Стоит отметить, что сервис Dialogs изначально писался, как самостоятельный сервис, потому шаг про «выделение функциональности» был здесь пропущен. Если я правильно уловил идею, то необходимо было реализовать проксирование запросов из основного сервиса (в данном проекте sso) к Dialogs, что сделано; хотя подход более чем странный, логичнее же организовать фасад Gateway, imho, чем мутировать одно из приложений в прокси...

Тестирование:

В коллекции postman переходим в папку Dialogs Obsolete и выполняем любой из запросов.

Переходим в логи контейнера sso и видим запись вида

```
Request started: 18d8d67e-6077-47b6-885c-b30b1677baa1
```

Переходим в логи контейнера dialogs, видим записи из middleware & repository вида

```
Request started: 18d8d67e-6077-47b6-885c-b30b1677baa1
```

```
18d8d67e-6077-47b6-885c-b30b1677baa1: Message de236b8a-dd6e-46a5-84e1-a61b378f710a sent from b90940ae-ae81-4f6c-b4f2-6986d0b91d4c to 1727f243-c1e3-46a9-9615-6a33c1fbf9e2
```

Видим, что id запросов (18d8d67e-6077-47b6-885c-b30b1677baa1) совпадают.