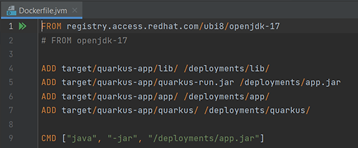
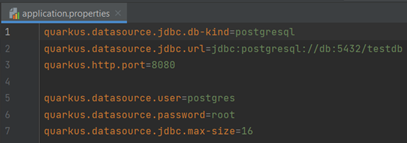
Создаем Dockerfile для сервиса.

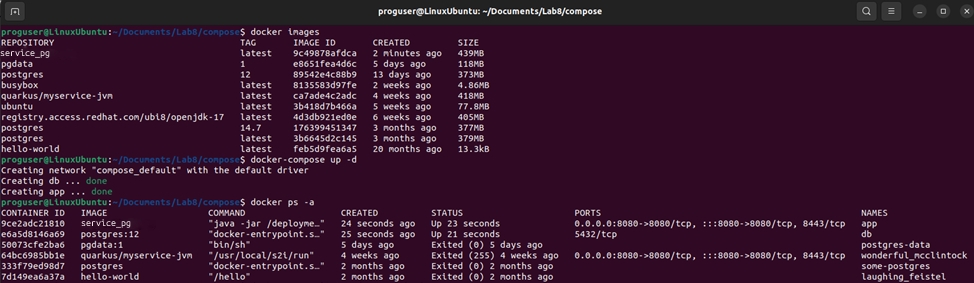


Прописываем настройки для подключения к БД в application.properties.

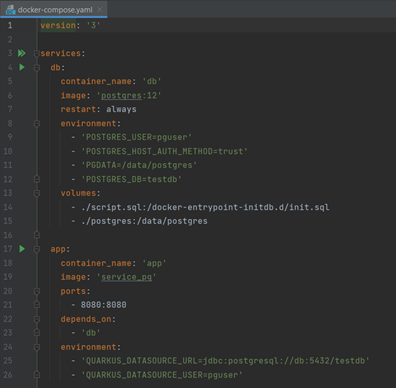
Создаем образ service\_pg с нашим сервисом с помощью команды:

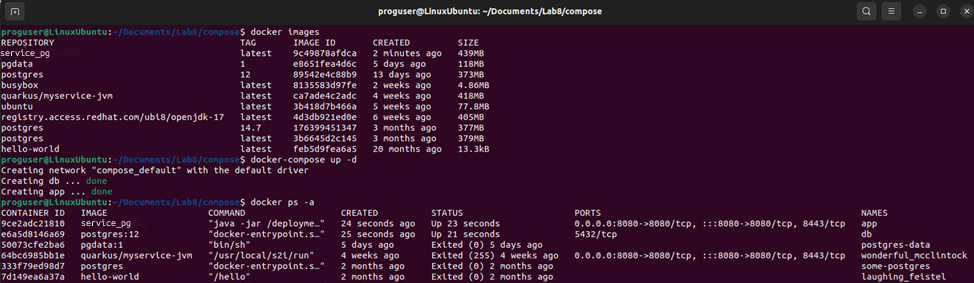
docker build -f src/main/docker/Dockerfile.jvm -t service\_pg .

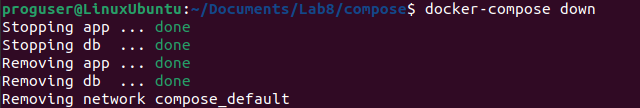
Проверяем его в списке образов.



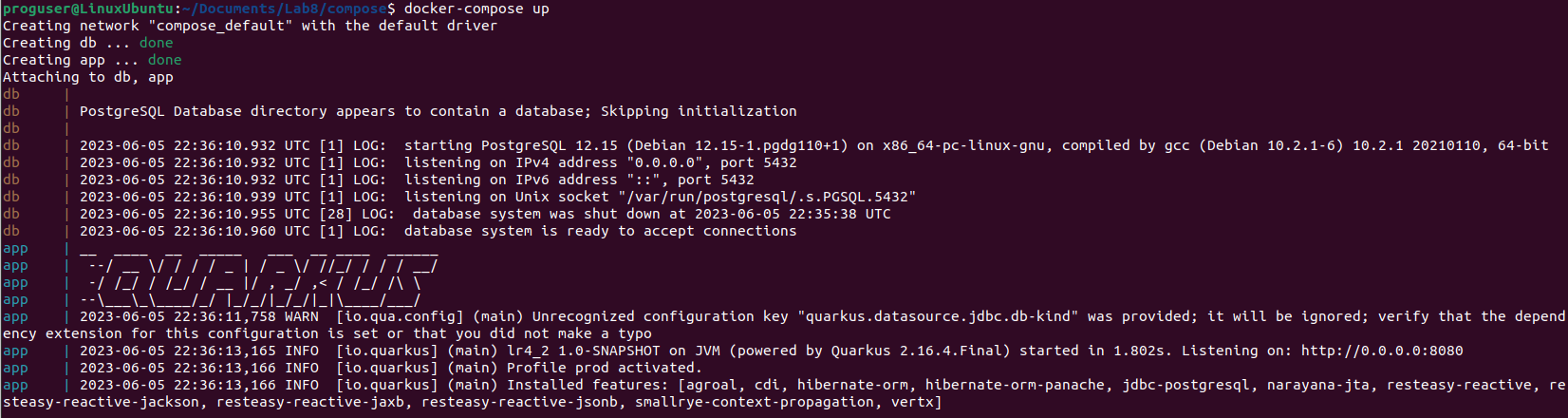
Создаем docker-compose.yaml файл для совместной работы двух контейнеров: с сервисом и с БД.

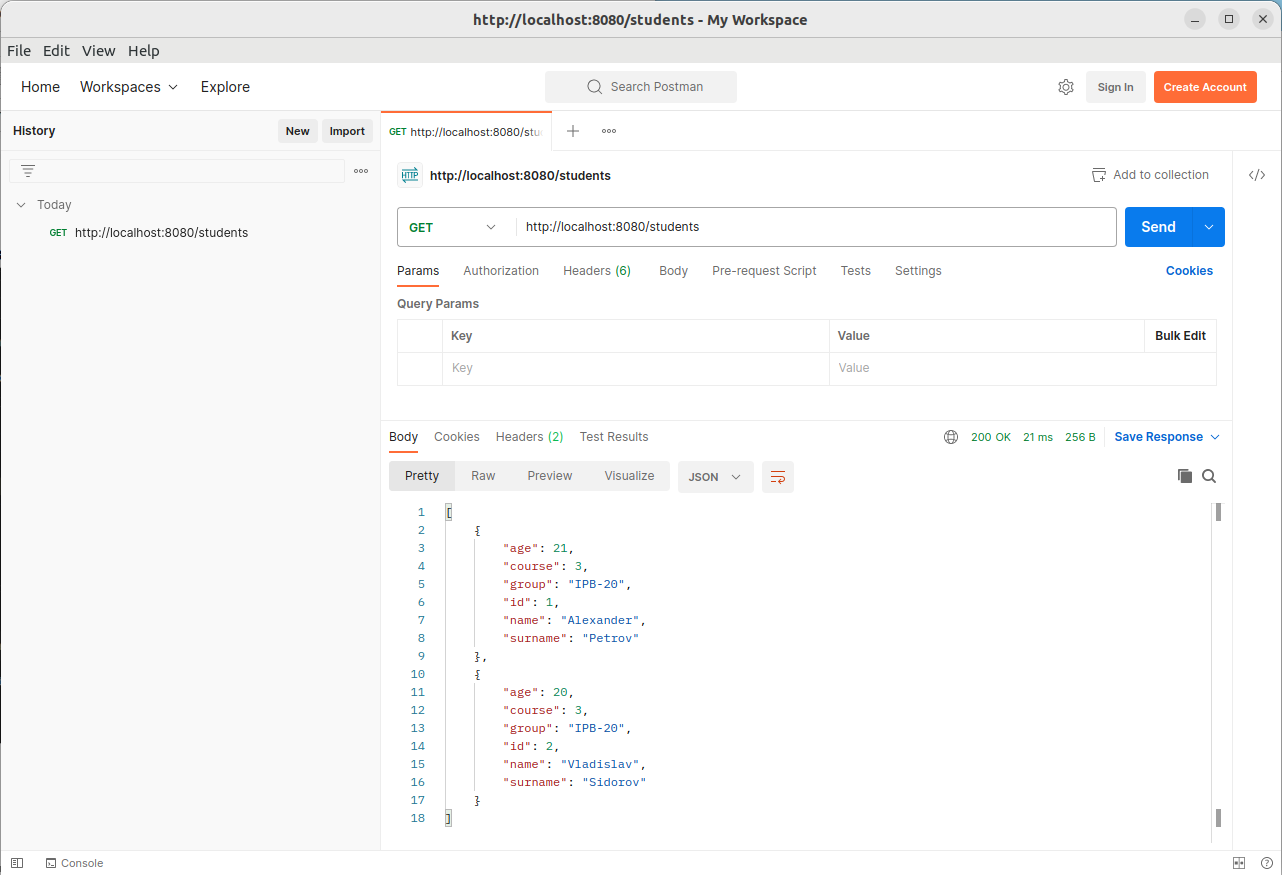
Также указываем в этом файле volumes, чтобы после удаления контейнеров все произведенные изменения сохранялись.

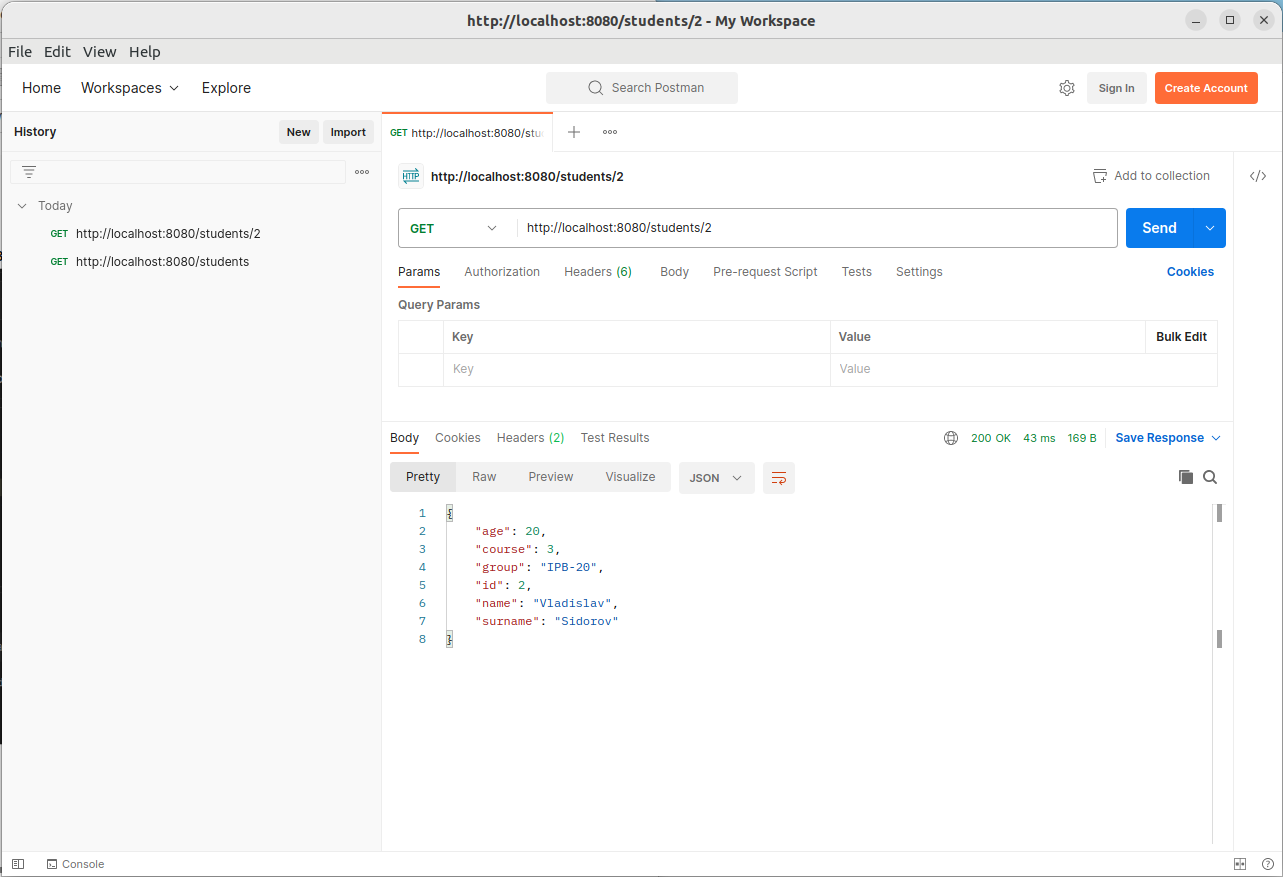
С помощью команды docker-compose up -d поднимаем контейнеры и проверяем с помощью команды docker ps -a.

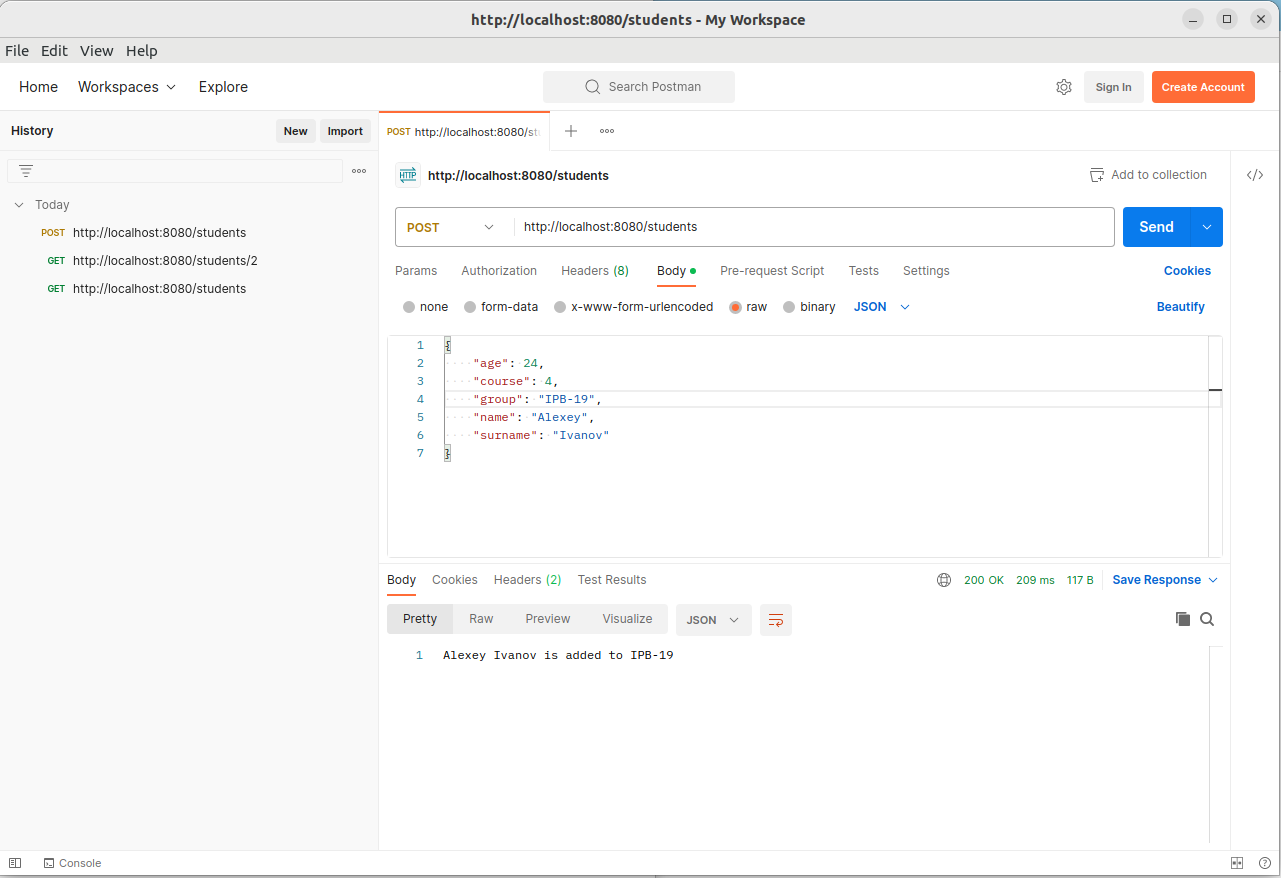


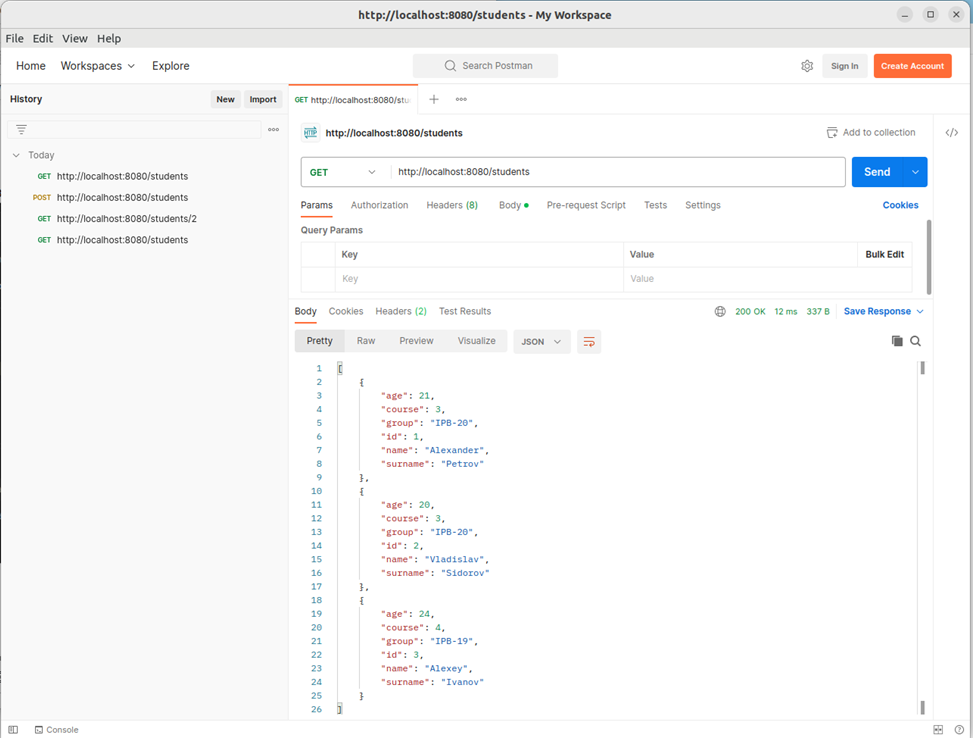
Запускаем контейнеры

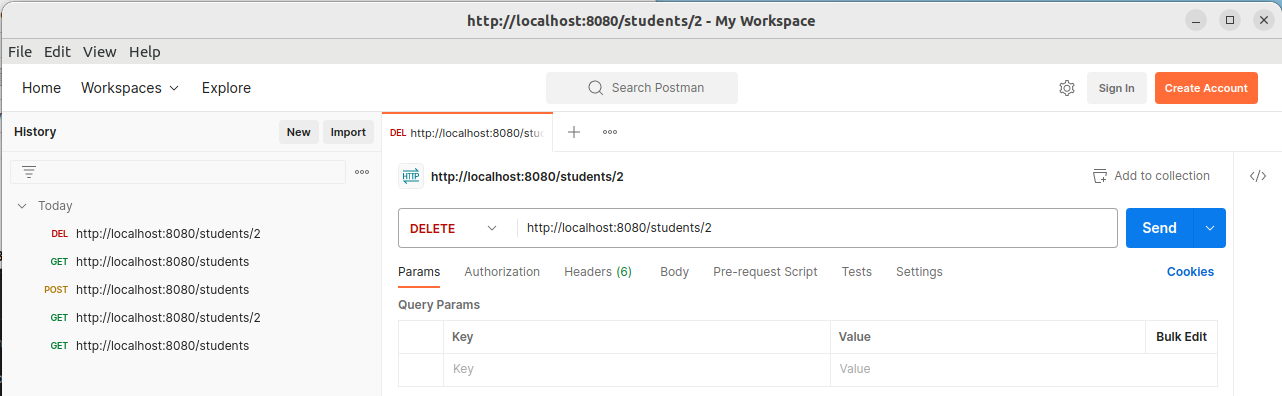


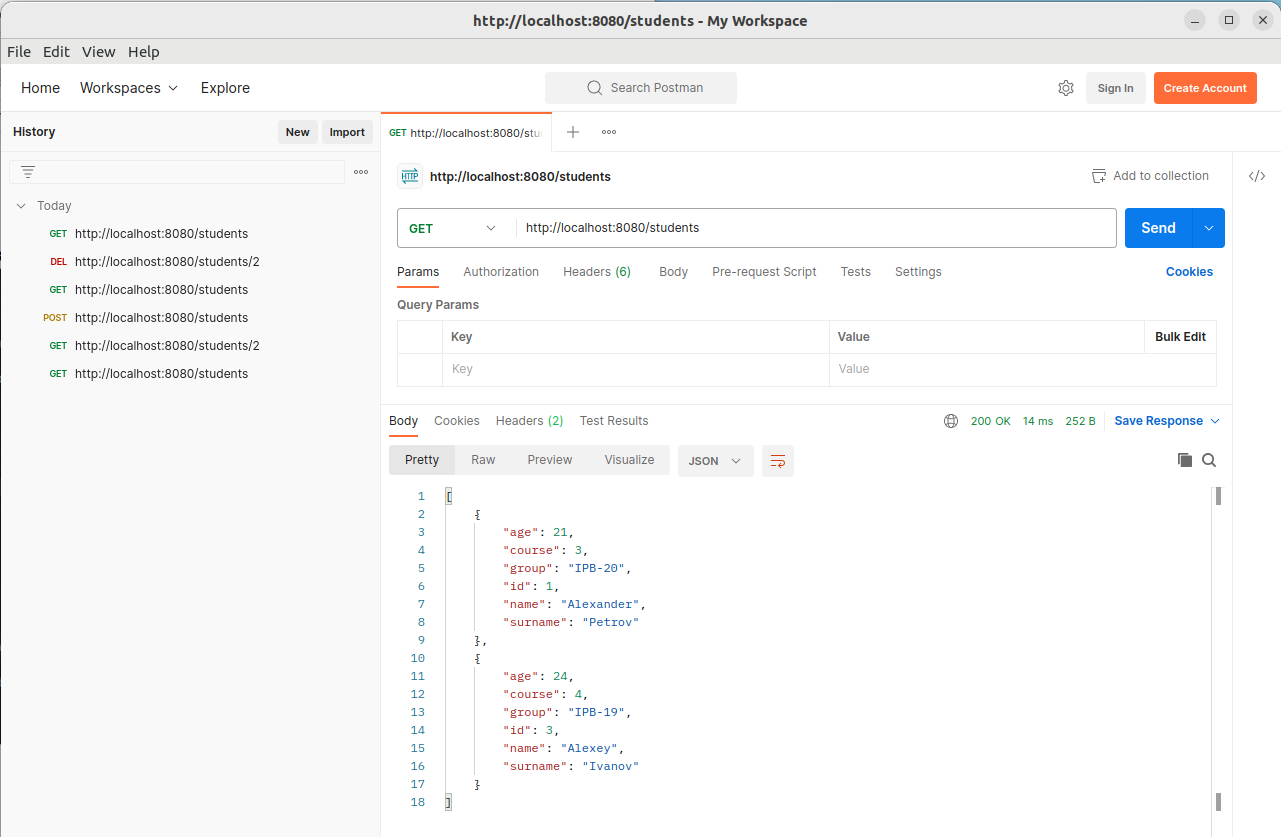
Проверим работоспособность, выведем список студентов:

Получим студента по id = 2:

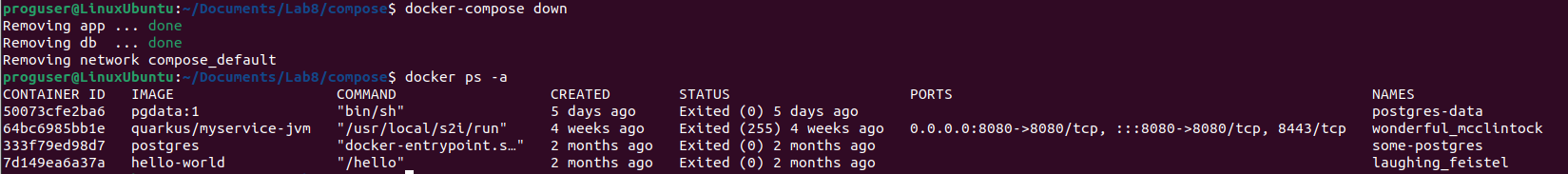
Добавим нового студента:

Выведем обновленный список всех студентов:

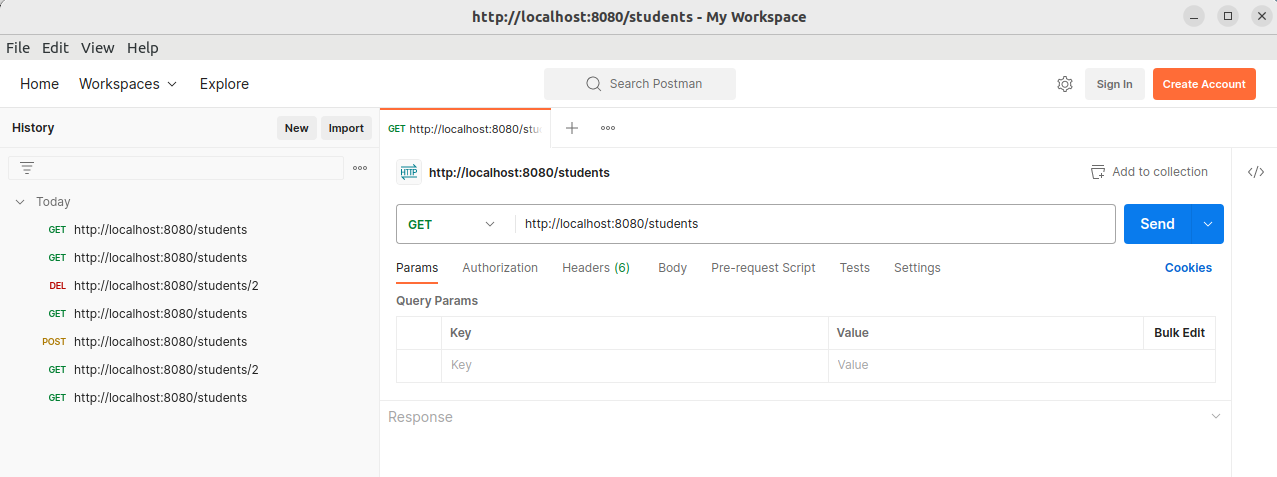
Удалим из списка студента с id = 2:

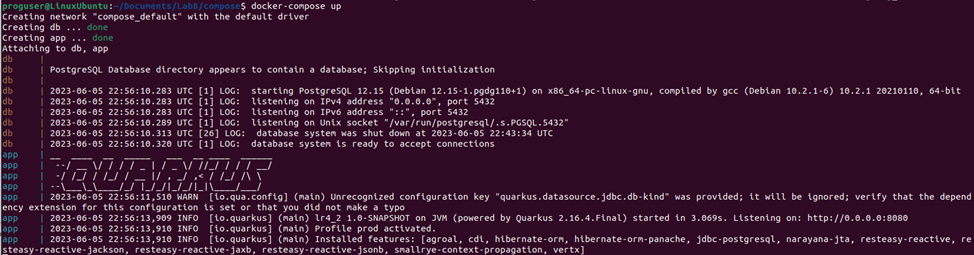


Прервем выполнение и удалим контейнеры с помощью команды docker-compose down и проверим, что они удалились:



После удаления контейнеров выполнение запросов прекратилось.



Поднимем контейнеры снова:

Снова запросим список студентов, остались с id = 1 и id = 3, а студент с id = 2 отсутствует в списке, т.е. все наши действия с предыдущего запуска docker-compose сохранились.

