|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №4**

по дисциплине «Разработка клиент-серверных приложений»

**Студент группы** ИКБО-20-19 Анваржонов Ж. Т

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель практической работы** преподаватель Волков М.Ю.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Москва 2022

**Создание реактивного приложения реального времени с использованием протокола Rsocket в JAVA**

**Задание:** Разработать клиент-серверную систему с использованием протокола RSocket. Предметная область разрабатываемой системы выбирается студентом самостоятельно, но должна быть уникальна в рамках учебной группы.

**Требования к сервису:**

- Должны быть продемонстрированы все 4 типа взаимодействия в рамках протокола RSocket:

o Request-Response;

o Request-Stream;

o Channel;

o Fire-and-Forget).

- Должна быть продемонстрирована работа с операторами преобразования потоков.

- Должно быть продемонстрировано взаимодействие с базой данных.

- Потоки должны закрываться после отработки бизнес-логики.

- Серверная часть приложения должна быть покрыта Unit-тестами

**Выполнение задания**

Продемонстрируем все 4 типа взаимодействия в рамкам протокола RSocket:

1. Request-Response



Рисунок 1 – клиентская часть обработки request-response



Рисунок 2 – обработки на стороне сервера request-response

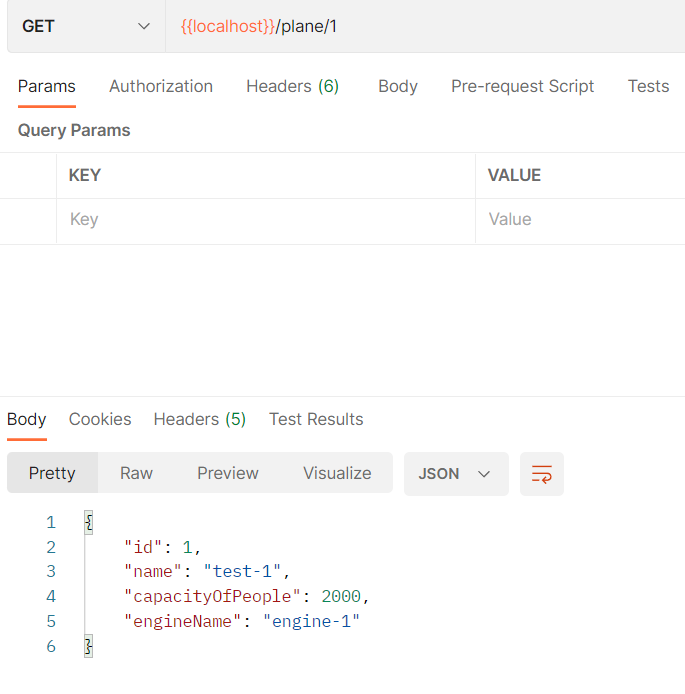


Рисунок 3 – демонстрация работы request-response

1. Request-Stream

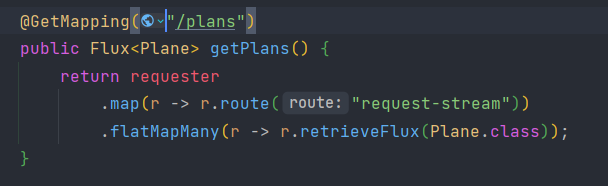


Рисунок 4 – клиентская часть обработки request-stream

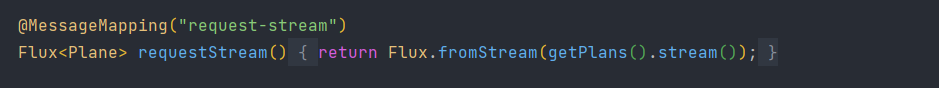


Рисунок 5 – обработки на стороне сервера request-stream

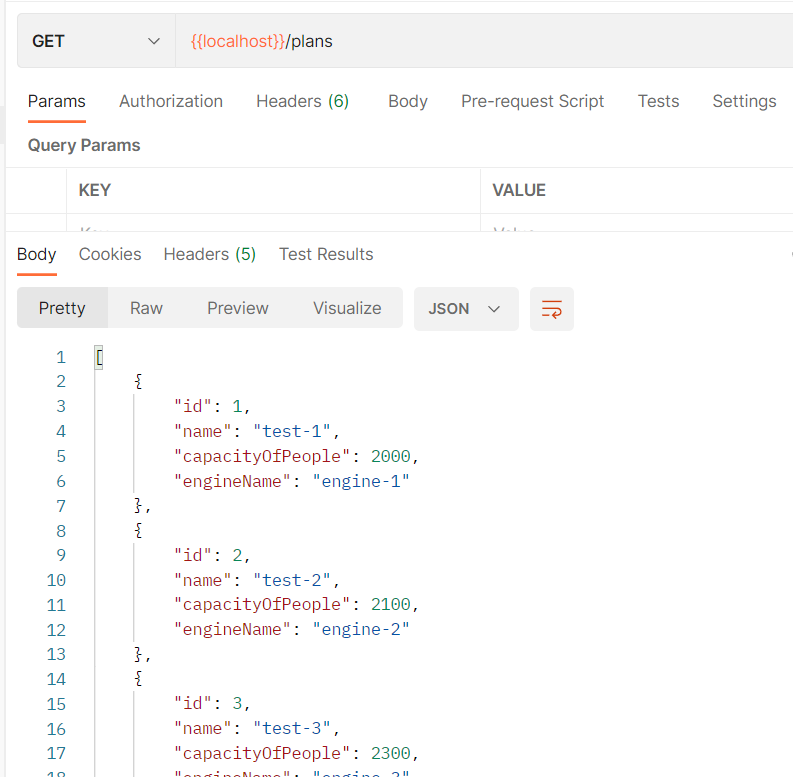


Рисунок 6 – демонстрация работы request-stream

1. Channel



Рисунок 7 – клиентская часть обработки channel

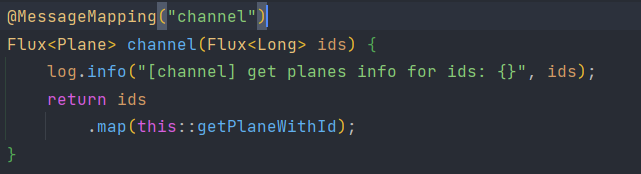


Рисунок 8 – обработки на стороне сервера channel

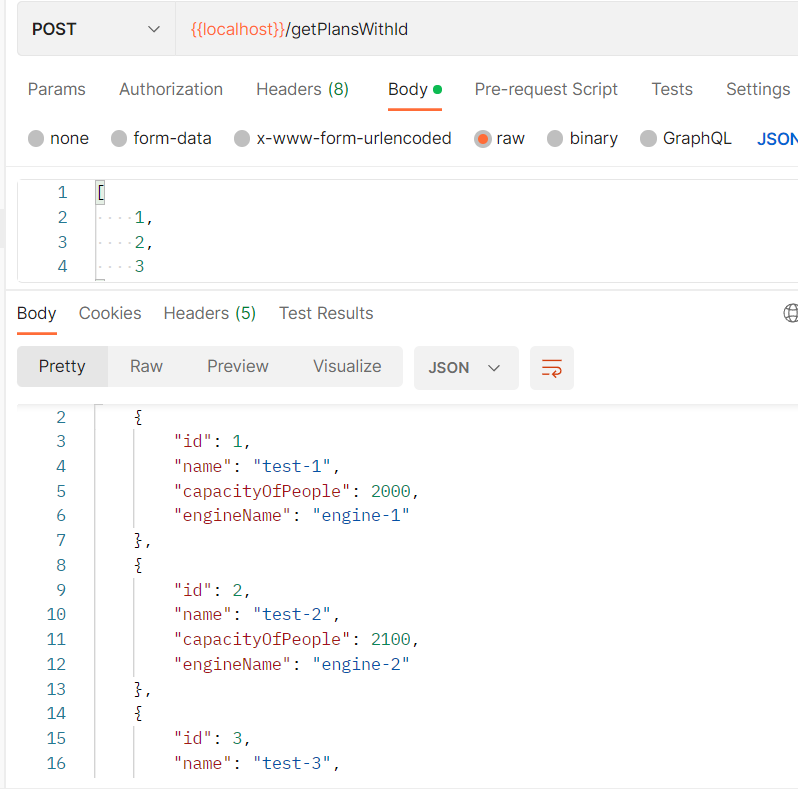


Рисунок 9 – демонстрация работы channel

1. Fire and Forget

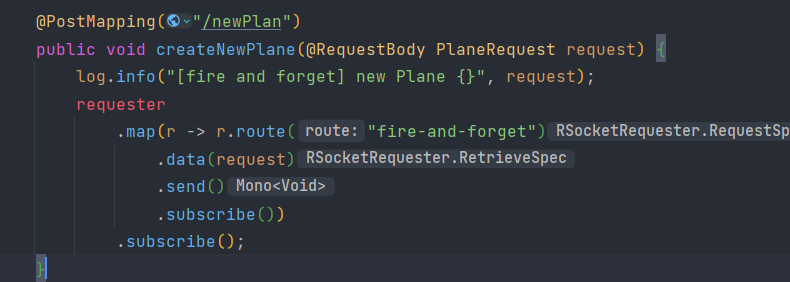


Рисунок 10 – клиентская часть обработки fire and forget

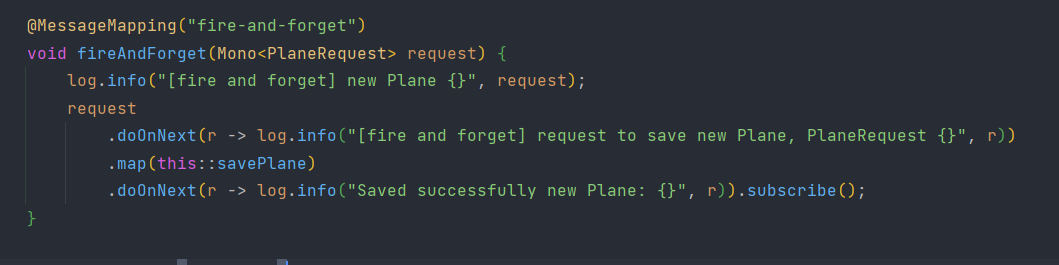


Рисунок 11 – Обработки на стороне сервера request-response

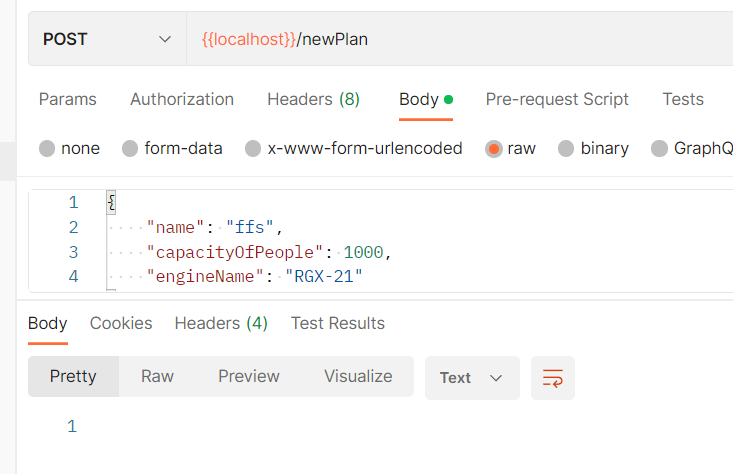


Рисунок 12 – Демонстрация работы fire-and-forget

Продемонстрируем работоспособность с базой данных postgresql

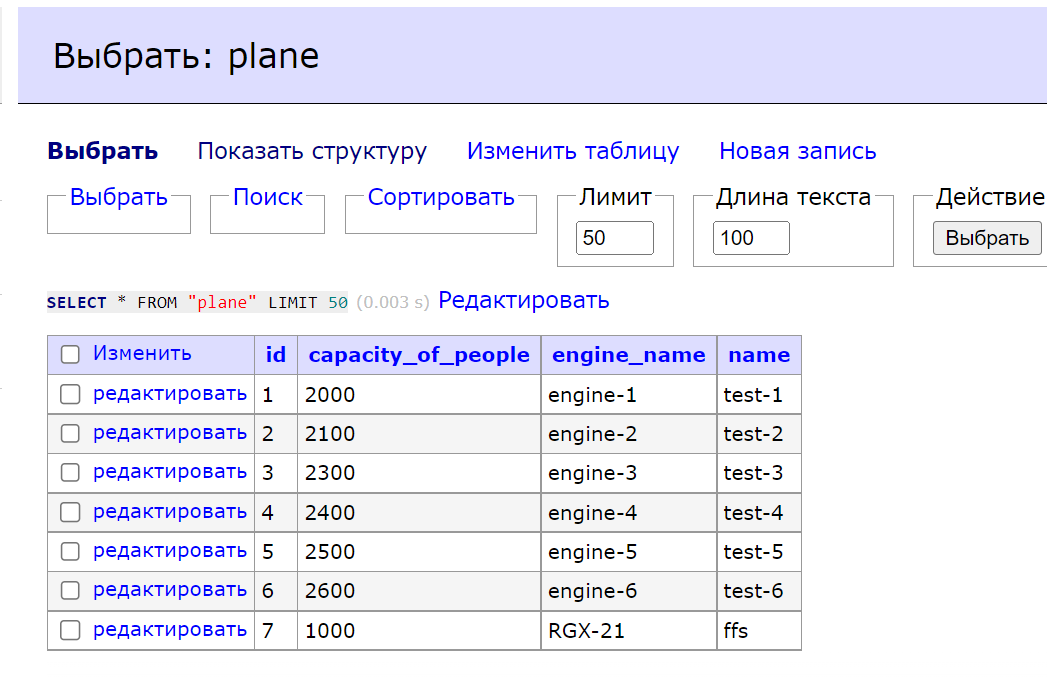


Рисунок 13 – База данных