

УНИВЕРСИТЕТ ИТМО

СЕРВИС ОРИЕНТИРОВАННАЯ АРХИТЕКТУРА

Лабораторная работа №3

ФИО студентов:

Готовко Алексей Владимирович

Руденко Илья Александрович

Вариант: 789

Направление подготовки: 09.03.04 (СППО)

Учебная группа: Р34101

ФИО преподавателя: Кривоносов Егор Дмитриевич

Санкт-Петербург

2024г.

Содержание

1	Задание	2
2	Инструкции по деплою	2
3	Исходный код	2
4	Вывод	2

1 Задание

Переработать веб-сервисы из лабораторной работы #2 таким образом, чтобы они реализовывали основные концепции микросервисной архитектуры. Для этого внести в оба сервиса – "вызываемый" и "вызывающий" – перечисленные ниже изменения.

Изменения в "вызываемом" сервисе:

- Сконфигурировать окружение для работы сервиса на платформе Spring Boot.
- Запустить второй экземпляр сервиса на другом порту. Реализовать балансировку нагрузки между экземплярами с помощью Nginx.
- Реализовать механизм Service Discovery. Для этого установить Consul и интегрировать свой сервис с ним, автоматически регистрируя в момент запуска.

Изменения в "вызывающем" сервисе:

- Разделить приложение на два модуля – веб-приложение с веб-сервисом и EJB-jar с бизнес-компонентами.
- Переместить всю логику из класса сервиса в Stateless EJB. В классе сервиса оставить только обращение к методам бизнес-интерфейса. EJB-компонент должен быть доступен удалённо (иметь Remote-интерфейс).
- Сформировать на уровне сервера приложений пул компонентов EJB настраиваемой мощности, динамически расширяемый при увеличении нагрузки.
- Настроить второй экземпляр сервера приложений на другом порту, "поднять" на нём вторую копию веб-сервиса и пула EJB.
- Настроить балансировку нагрузки на оба запущенных узла через Nginx.

Оба веб-сервиса и клиентское приложение должны сохранить полную совместимость с API, реализованными в рамках предыдущих лабораторных работ.

2 Инструкции по деплою

В качестве дополнительного задания необходимо использовать Docker для развертывания приложений.

Инструкции по деплою доступны по [ссылке](#).

(<https://github.com/xGodness/service-oriented-architecture/blob/master/README.md>).

3 Исходный код

Доступен по [ссылке](#).

(<https://github.com/xGodness/service-oriented-architecture>).

4 Вывод

В результате выполнения лабораторной работы были в достаточной степени освоены принципы микросервисной архитектуры, а также получен опыт работы с Consul, Nginx и технологией EJB. Было получено неимоверное удовольствие.