

Национальный исследовательский университет ИТМО
Факультет Программной Инженерии и Компьютерной Техники



Лабораторная работа 3
по дисциплине
“Сервис-ориентированная архитектура”
Вариант 10134

Выполнил:
Студент группы Р34101
Иннокентьев Артем Алексеевич
Преподаватель:
Цопа Евгений Алексеевич

Санкт-Петербург
2024

Текст задания

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.

Выполнение

Ссылка на Github:

Сервис 1: https://github.com/artemychych/soa2024_service1

Сервис 2 EJB: https://github.com/artemychych/soa2024_ejb

Сервис 2 Backend: https://github.com/artemychych/soa2024_backend

Вывод

Во время выполнения лабораторной работы я реализовал балансировку нагрузки между экземплярами вызываемых сервисов с помощью Nginx, реализовал механизм Service Discovery с помощью Consul. Разделил вызывающий сервис на EJB-jar и веб-сервис на JAX-RS, настроил балансировку нагрузки с помощью Nginx.