

НИУ ИТМО

Факультет программной инженерии и компьютерных технологий

Отчет по лабораторной работе №3

по дисциплине Бизнес-логика программных систем

Студент группы № Р33151

Шипулин Павел Андреевич

Преподаватель

Кривоносов Егор Дмитриевич

Санкт-Петербург

2024

Задание (вариант 9177)

Доработать приложение из лабораторной работы #2, реализовав в нём асинхронное выполнение задач с распределением бизнес-логики между несколькими вычислительными узлами и выполнением периодических операций с использованием планировщика задач.

Требования к реализации асинхронной обработки:

1. Перед выполнением работы необходимо согласовать с преподавателем набор прецедентов, в реализации которых целесообразно использование асинхронного распределённого выполнения задач. Если таких прецедентов использования в имеющейся бизнес-процесса нет, нужно согласовать реализацию новых прецедентов, доработав таким образом модель бизнес-процесса из лабораторной работы #1.
2. Асинхронное выполнение задач должно использовать модель доставки "подписка".
3. В качестве провайдера сервиса асинхронного обмена сообщениями необходимо использовать сервис подписки на базе Apache Kafka + ZooKeeper.
4. Для отправки сообщений необходимо использовать Kafka Producer API.
5. Для получения сообщений необходимо использовать клиент Kafka на базе Spring Boot.

Требования к реализации распределённой обработки:

1. Обработка сообщений должна осуществляться на двух независимых друг от друга узлах сервера приложений.

2. Если логика сценария распределённой обработки предполагает транзакционность выполняемых операций, они должны быть включены в состав распределённой транзакции.

Требования к реализации запуска периодических задач по расписанию:

1. Согласовать с преподавателем прецедент или прецеденты, в рамках которых выглядит целесообразным использовать планировщик задач. Если такие прецеденты отсутствуют -- согласовать с преподавателем новые и добавить их в модель автоматизируемого бизнес-процесса.
2. Реализовать утверждённые прецеденты с использованием планировщика задач Quartz.

Правила выполнения работы:

1. Все изменения, внесённые в реализуемый бизнес-процесс, должны быть учтены в описывающей его модели, REST API и наборе скриптов для тестирования публичных интерфейсов модуля.
2. Доработанное приложение необходимо либо развернуть на сервере helios, либо продемонстрировать его работоспособность на собственной инфраструктуре обучающегося.

Содержание отчёта:

1. Текст задания.
2. Модель потока управления для автоматизируемого бизнес-процесса со всеми внесёнными изменениями.
3. UML-диаграммы классов и пакетов разработанного приложения.
4. Спецификация REST API для всех публичных интерфейсов разработанного приложения.

5. Исходный код системы или ссылка на репозиторий с исходным кодом.
6. Выводы по работе.

Модель потока управления

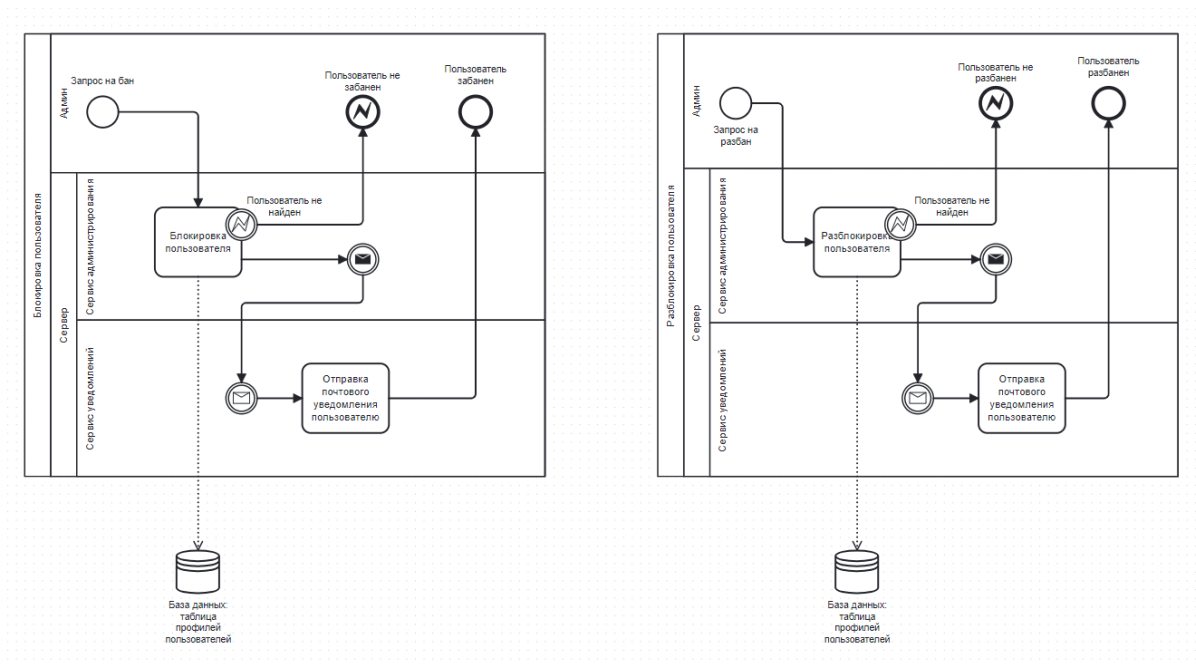


Рисунок 1. Изменения в модели процесса.

https://github.com/PashcalE2/BLPS/tree/main/BLPS_lab3/report/diagram.svg

UML-диаграммы классов и пакетов



https://github.com/PashcalE2/BLPS/tree/main/BLPS_lab3/report/blps_lab3.png

Спецификация REST API

Изменений в спецификации REST API по сравнению с интерфейсами в лабораторной работе №2 – нет.

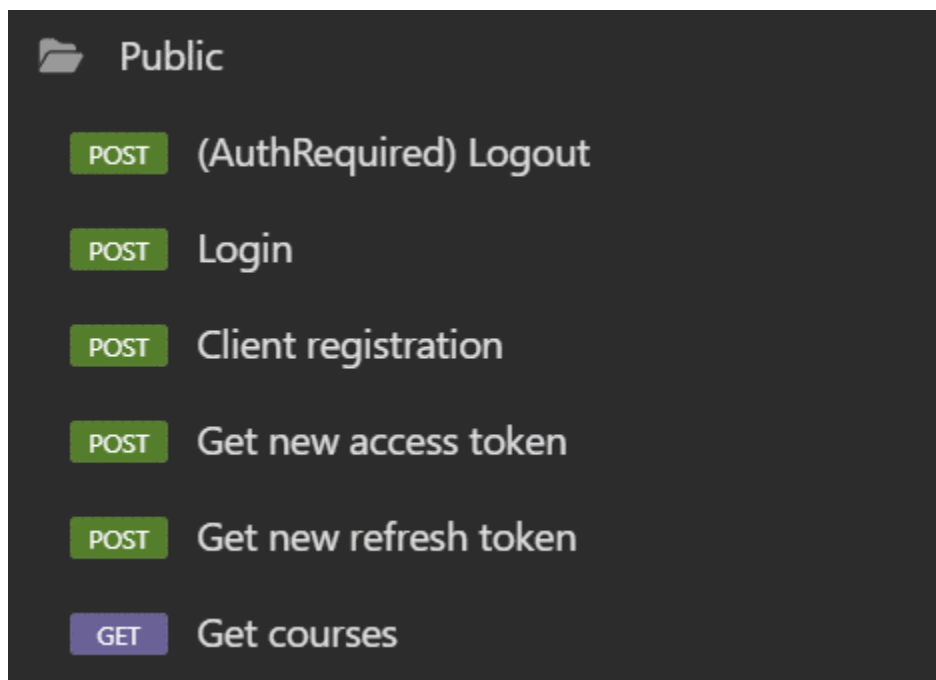


Рисунок 2. Набор публичных API.

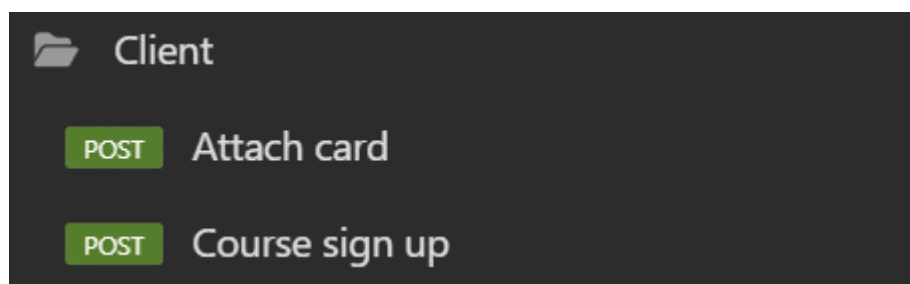


Рисунок 3. Набор API для пользователей с ролью “Клиент”

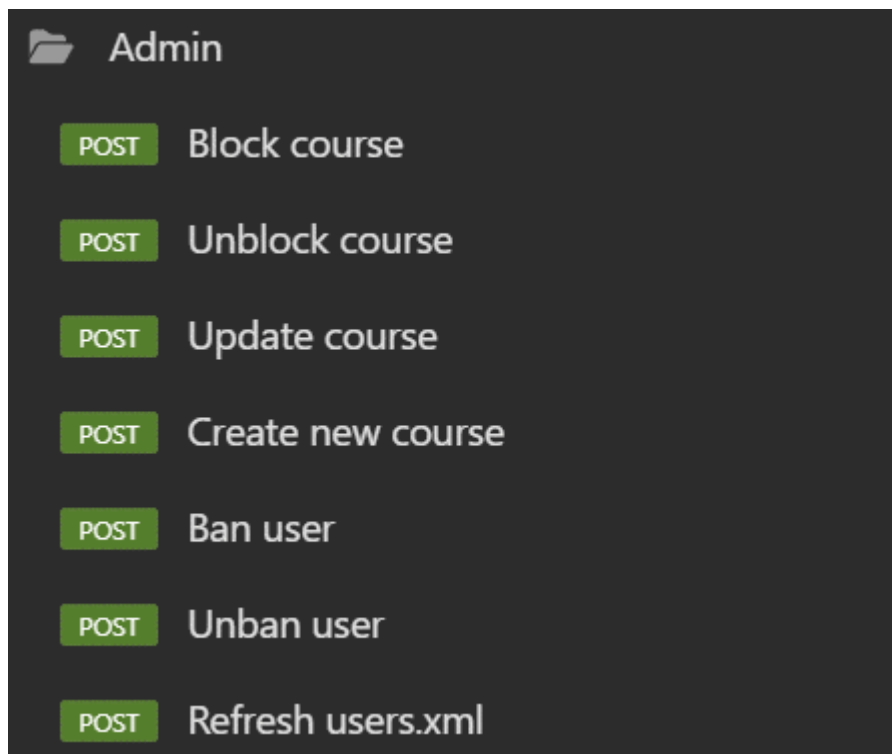


Рисунок 4. Набор API для пользователей с ролью “Администратор”

Исходный код

https://github.com/PashcalE2/BLPS/tree/main/BLPS_lab3

Выводы

Использовал декларативное управление транзакциями в требующих этого прецедентах.

Научился организовывать модели разграничения доступа на базе Spring Security. (He) Интегрировал JAAS по официальной документации Spring. Реализовал JWT токены (refresh и access) для аутентификации пользователей после первичной проверки логина и пароля.