УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Бизнес логика программных систем»

Лабораторная работа №3

Вариант 1209

Студент

Патутин Владимир

Иннокентьев Артем

P33101

Преподаватель

Цопа Е. А.

Санкт-Петербург, 2022 г.

Задание лабораторной работы

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 неободимо 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 "очередь сообщений".
3. В качестве провайдера сервиса асинхронного обмена сообщениями необходимо использовать очередь сообщений на базе RabbitMQ.
4. Для отправки сообщений необходимо использовать протокол AMQP 0-9-1. Библиотеку для реализации отправки сообщений можно взять любую на выбор студента.
5. Для получения сообщений необходимо использовать слушателя сообщений JMS на базе Spring Boot (@JmsListener).

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.

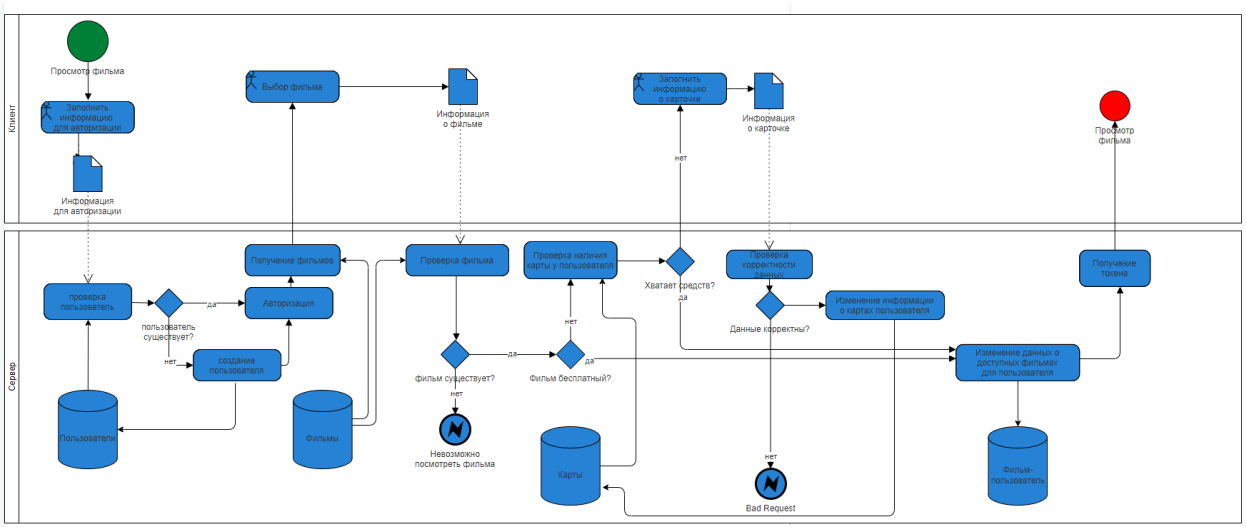
Требования к реализации запуска периодических задач по расписанию:

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 Spring (@Scheduled).

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.

Модель бизнес-процесса «Редактирование и проверка статьи»

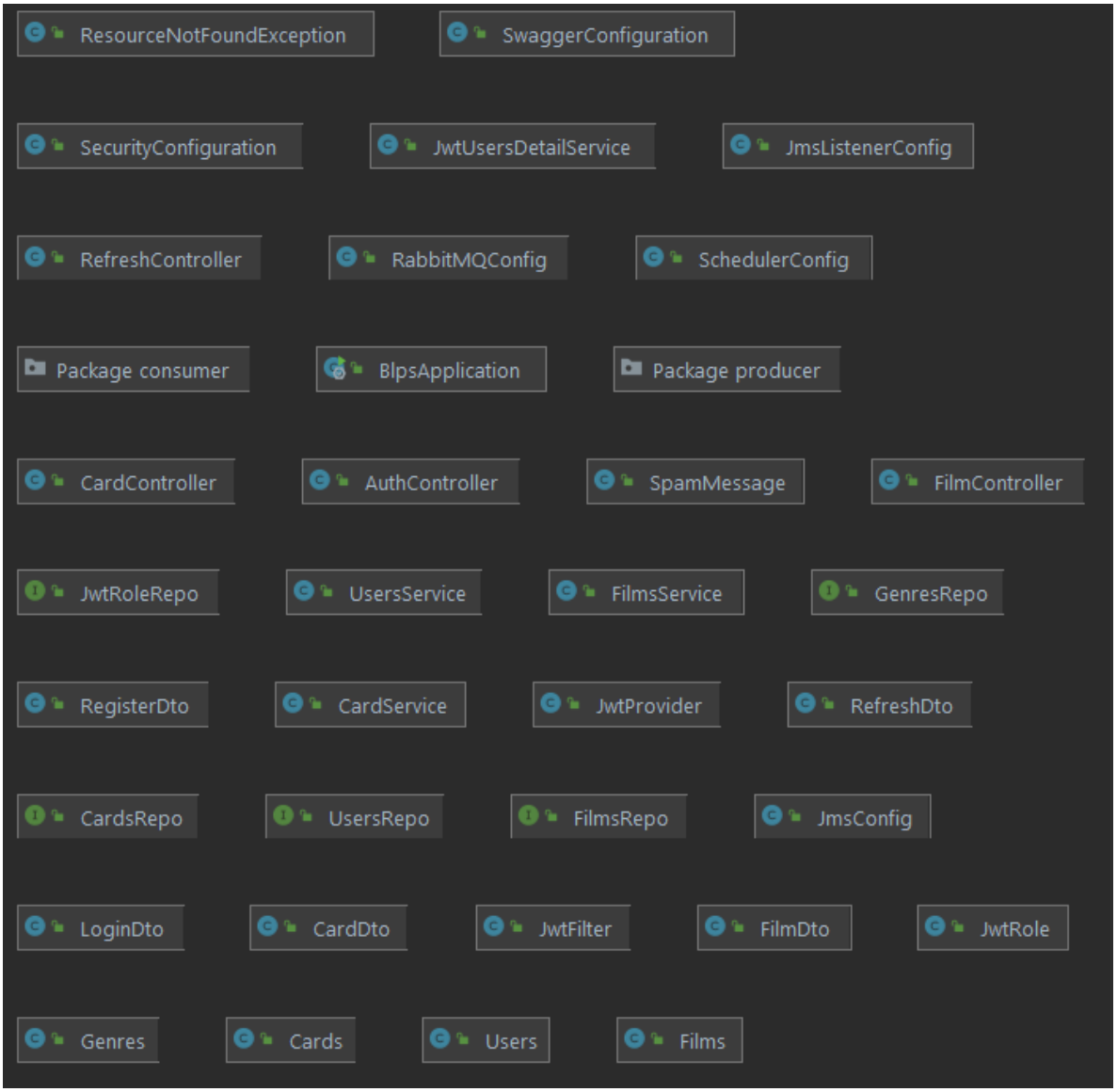


Спецификация пользовательский привилегий и ролей

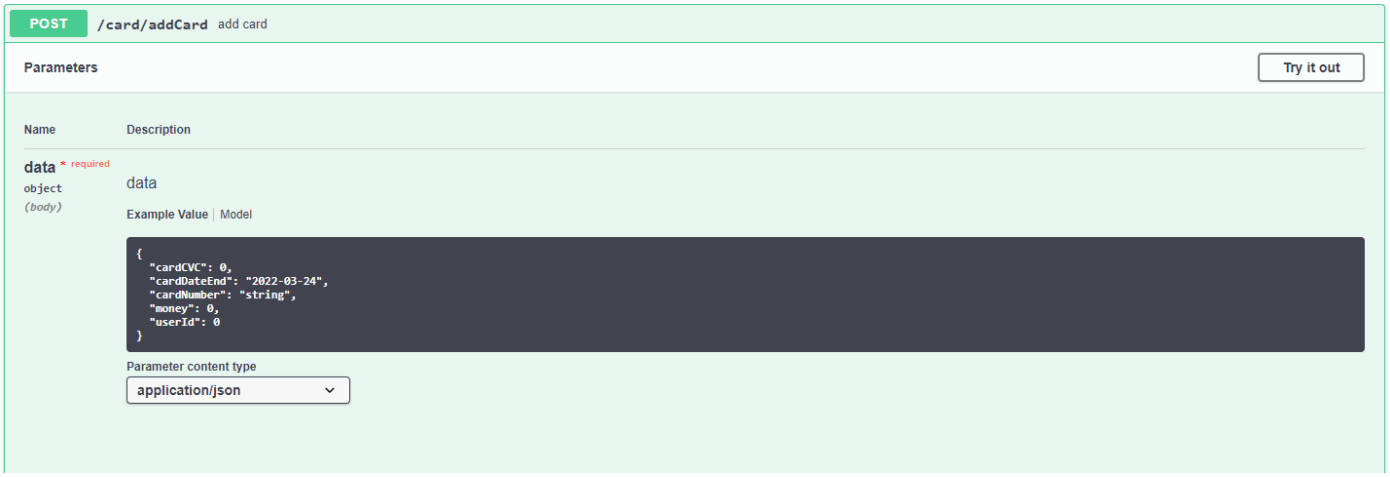
В разработанном приложении реализована одна роль –пользователь.

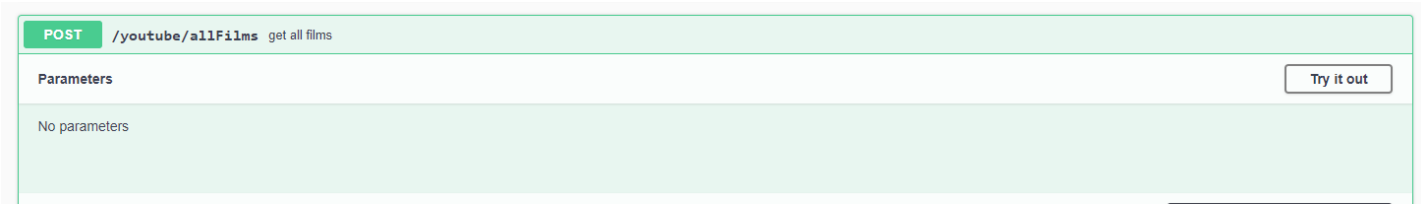
Пользователь имеет доступ к просмотру фильмов и их покупке

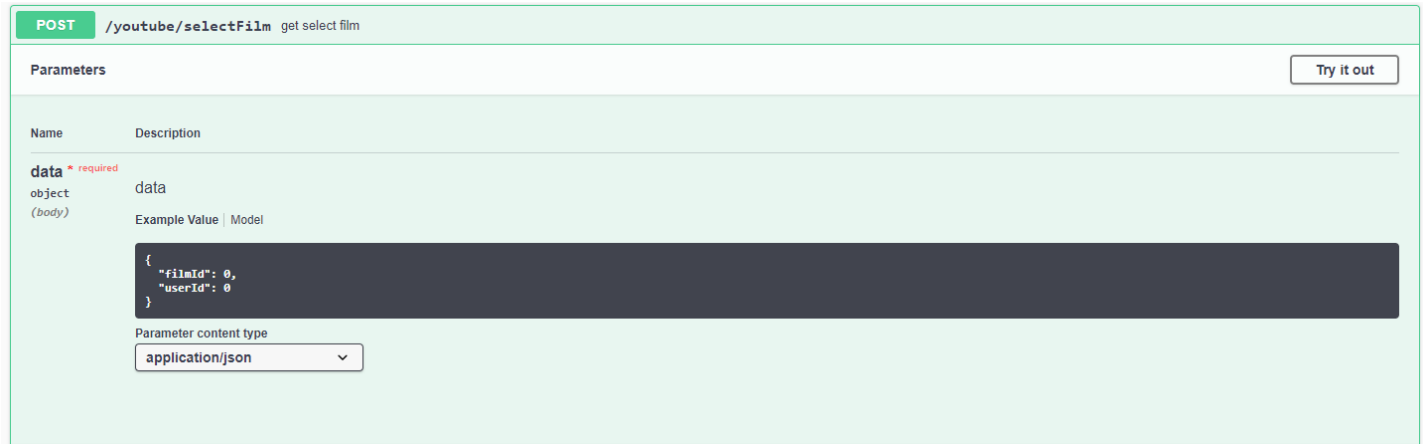
UML-Диаграмма классов



Спецификация REST API разработанного приложения







Исходный код

https://github.com/DeltaHeavyVIP/Business-logic-software-systems

Вывод

Таким образома, в процессе выполнения данной лабараторной, наше приложение перестало быть монолитным и было разделено на 2 отдельных, независимых друг от друга узла, работающих на сервере приложений WildFLy. Взаимодействие между которыми осуществляет при помощи сервиса асинхронного обмена сообщениями RabbitMQ. Также был добавлен новый прецедент использования, который использует функционал запуска периодических задач по расписанию при помощи аннотации @Scheduled. REST API, не был изменен, поскольку модификации касались только внутреннего устройства сервиса.