**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное учреждение

высшего образования «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ

Центр ускоренного обучения

Отчет по лабораторной работе № 4

по дисциплине «Технологии Программирования»

Старший преподаватель: Н.А. Архипов

Студент гр. РИВ-220938у М.С. Ермаков

Екатеринбург 2023

**1. Цель работы:**

**Цель:** Доработать простой REST сервис

**Учебные вопросы:**

1) Создание копии сервиса (далее Сервис 2), реализованного в ЛР 2 и ЛР 3 (далее Сервис 1), который буlет принимать request от Сервиса 1 доработанного в текущей ЛР

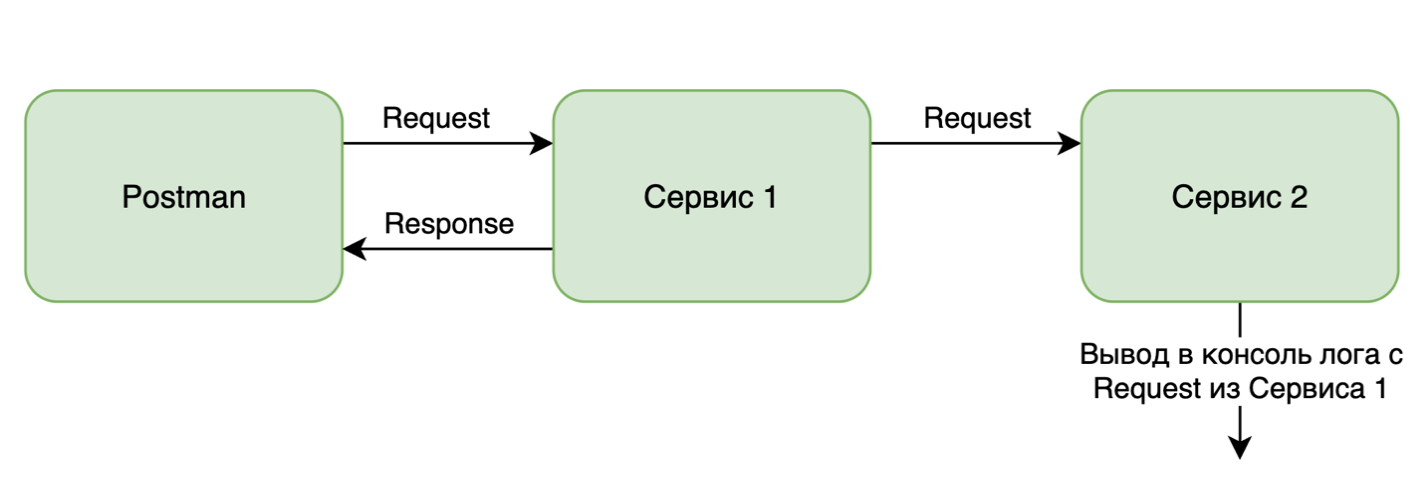
2) Доработка Сервиса 1 для того, чтобы он мог пересылать полученный и модифицированный request.

3) Ответить письменно на вопросы в соответствии с номером в списке

4) Оформить отчет и прикрепить его на сайте https://edu.itlearn.ru/ в соответствующем курсе, в соответствующем разделе

1.1 Схема работы сервисов и описание задачи

1.2 Схема работы сервисов



**2. Описание задачи:**

Реализация дополнительного функционала

Необходимо самостоятельно доработать приложение:

1. Добавьте реализацию интерфейса ModifyRequestService, которая изменяет поле source в Сервисе 1

2. Доработайте Сервис 1 и Сервис 2 таким образом, чтобы в логи Сервиса 2 выводилось сообщение о разнице времени прошедшего с момента получения Сервисом 1 Request и временем получения модифицированного Request полученного Сервисом 2, другими словами изерить время от нажатия кнопки Send в Postman до вывода в лог Сервисом 2 сообщения.

Вопросы

Письменно ответьте на вопросы, если ваш номер в списке группы четный, то нужно ответить на четные вопросы, если нечетный, то нужно ответить на нечетные вопросы.

1. Зачем нужен параметр server.port?

2. Зачем нужно аннотация @SpringBootApplication?

3. Зачем нужна аннотация @Builder?

4. Зачем нужна аннотация @NoArgsConstructor?

5. Зачем нужна аннотация @AllArgsConstructor?

6. Зачем нужна аннотация @Data?

**3. Ход выполнения:**

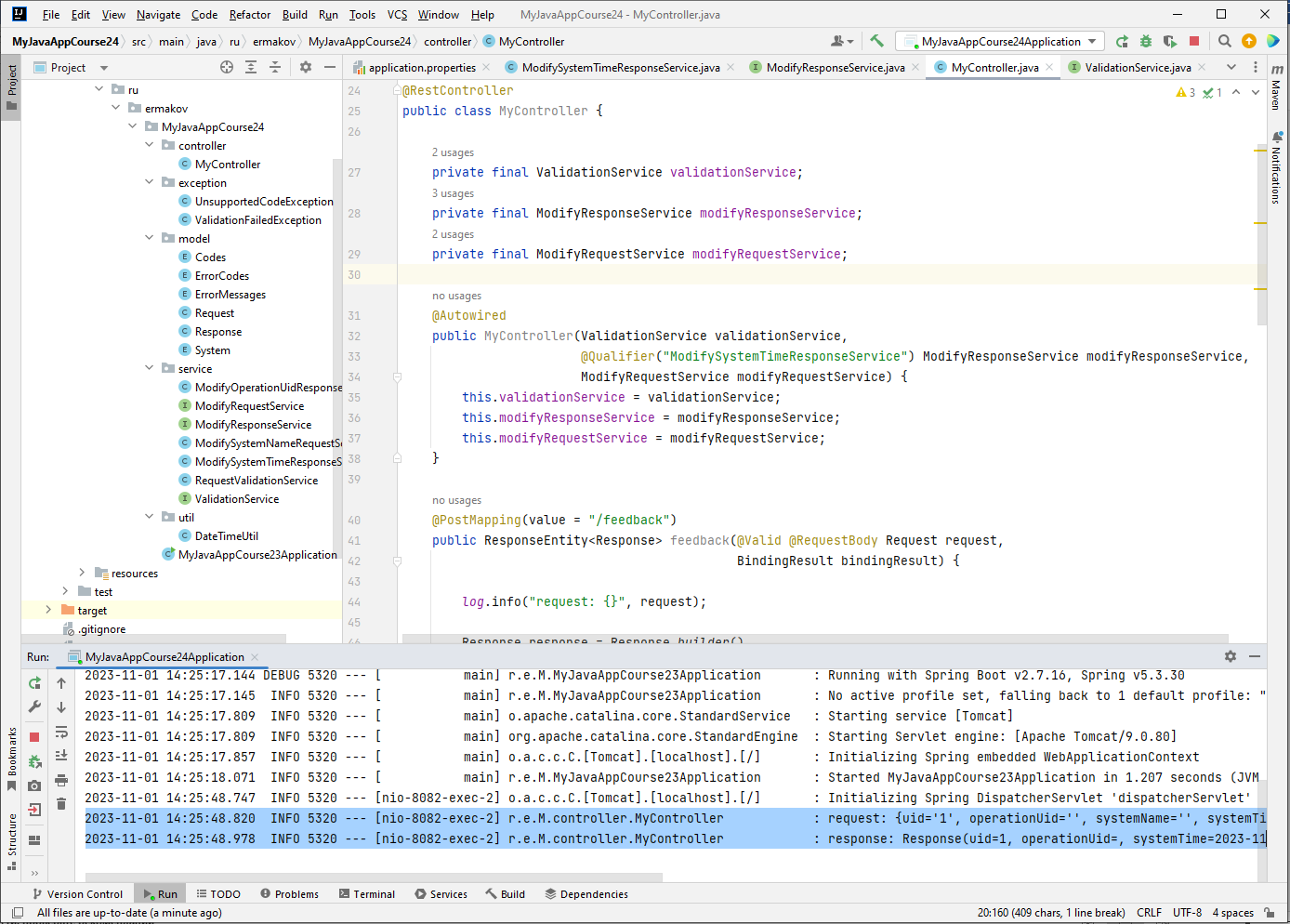


Рисунок 1 – Структура и логи Основного сервиса после POST в Postman

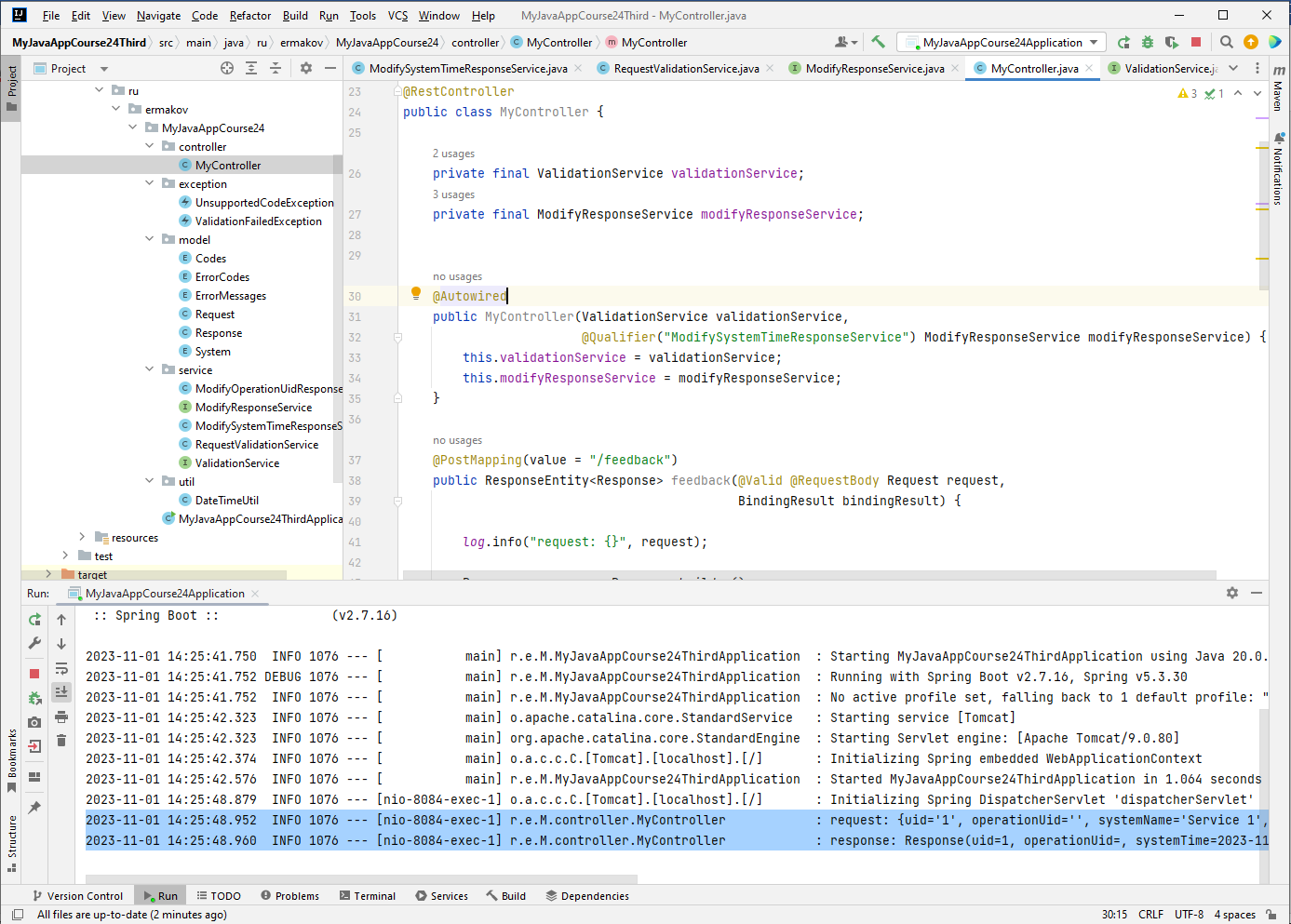


Рисунок 2 – Структура и логи Дополнительного сервиса Third после POST в Postman

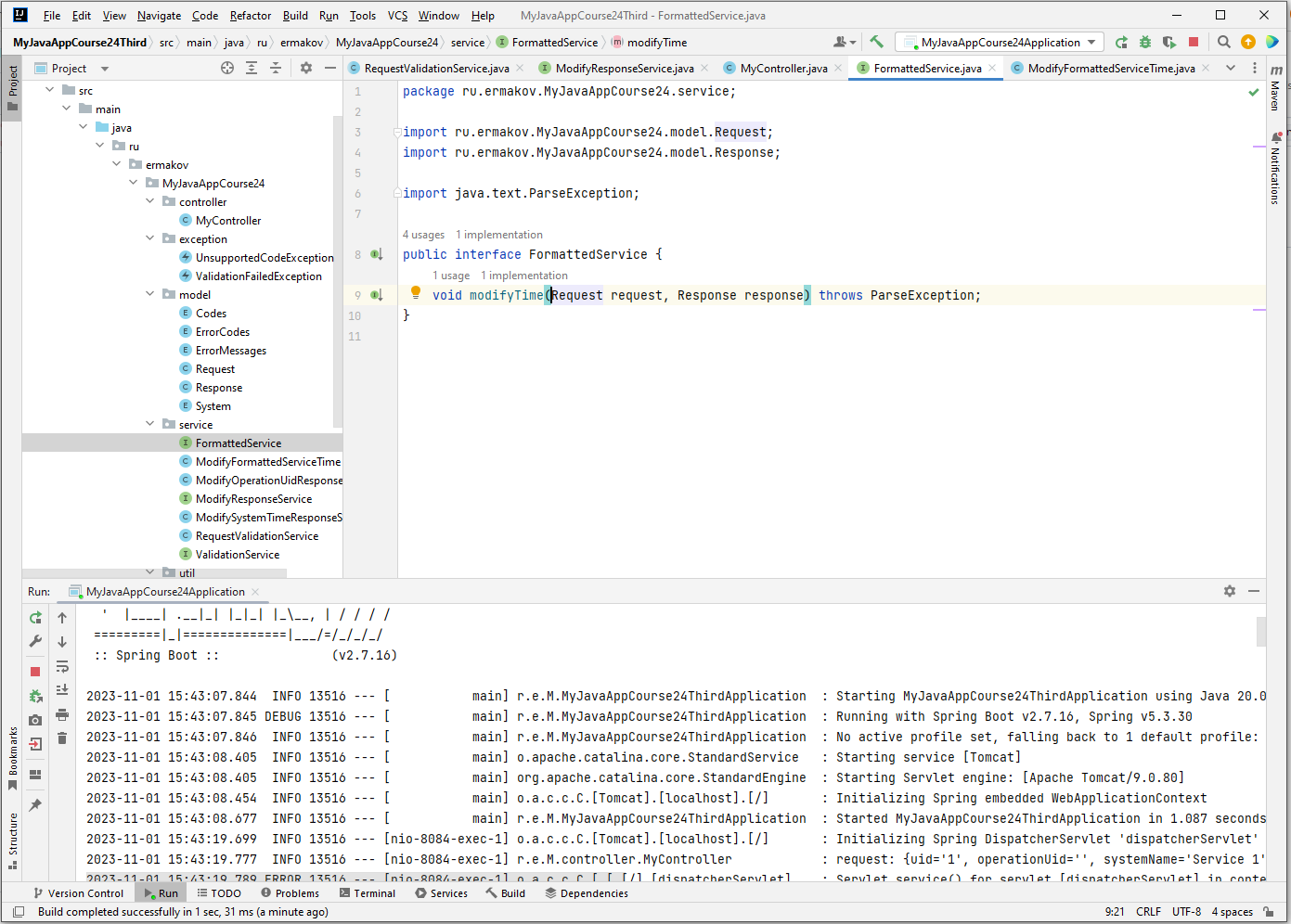


Рисунок 3 – Добавление интерфейса FormattedService для отслеживания времени

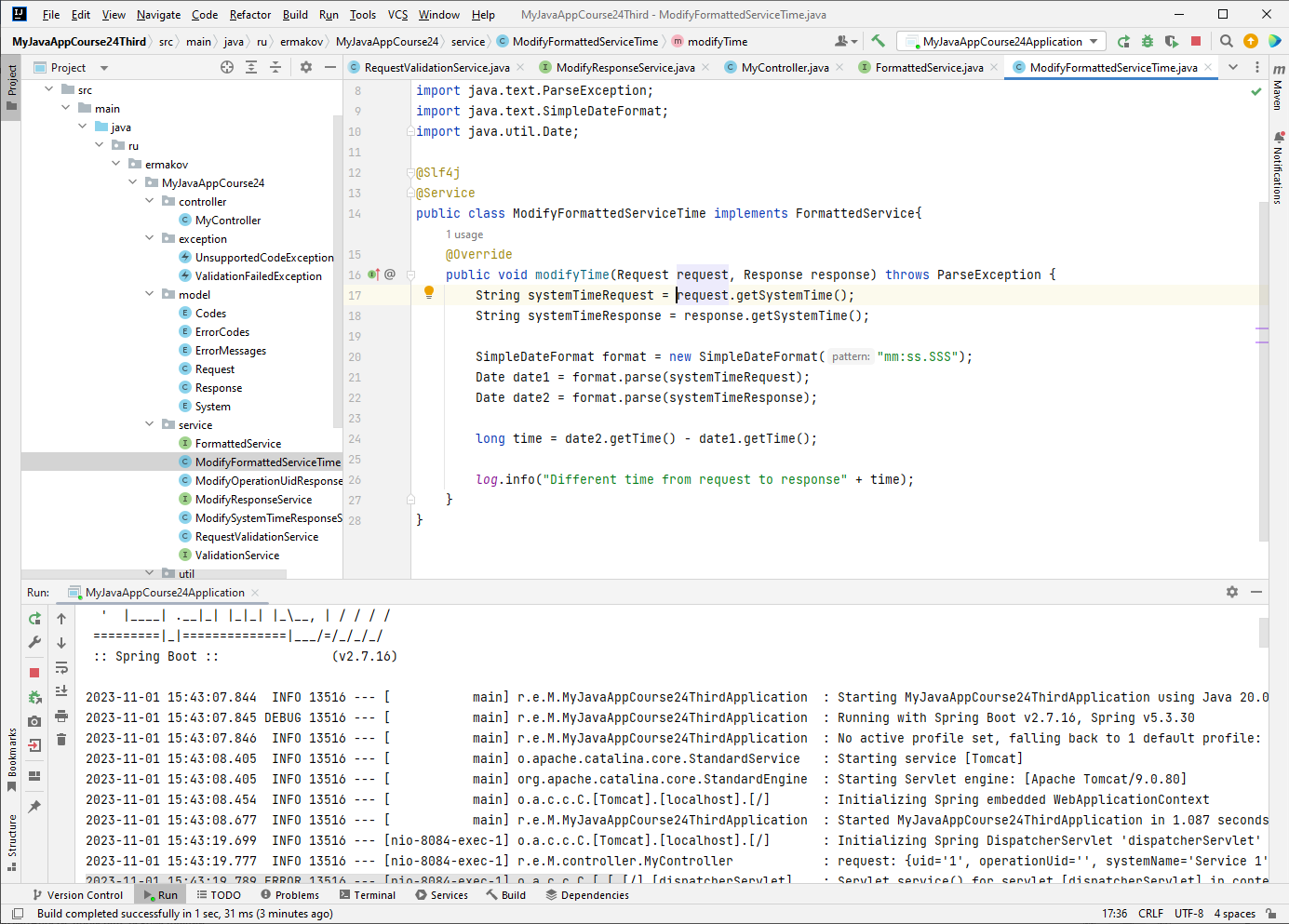


Рисунок 4 – Реализация интерфейса FormattedService для отслеживания времени

**Ответы на вопросы:**

2) Зачем нужно аннотация **@SpringBootApplication**?

Одну аннотацию **@SpringBootApplication** можно использовать для включения этих трех функций, а именно:

**@EnableAutoConfiguration**: включить механизм автоконфигурации Spring Boot

**@ComponentScan**: включить сканирование @Component для пакета, в котором находится приложение

**@Configuration**: позволяет регистрировать дополнительные компоненты (beans) в контексте или импортировать дополнительные классы конфигурации

SpringBootApplication также предоставляет псевдонимы для настройки атрибутов @EnableAutoConfiguration и @ComponentScan.

Ни одна из этих функций не является обязательной, и мы можем заменить эту единственную аннотацию любой из функций, которые она включает. Например, мы можем не захотеть использовать сканирование компонентов или сканирование свойств конфигурации в вашем приложении

4. Зачем нужна аннотация **@NoArgsConstructor**?

Это аннотация в библиотеке – Lombok.

**@NoArgsConstructor** — используется для создания конструктора без аргументов.

6) Зачем нужна аннотация **@Data**?

**@Data** – объединяет аннотации **@ToString**, **@Getter**, **@Setter**, **@EqualsAndHashCode** и **@RequiredArgsConstructor** в одну. Предоставляет весь код, который обычно используется в классах моделей, например, геттеры для всех полей, сеттеры для всех не финальных полей, реализацию по умолчанию для **toString()**, **equals()** и **hashCode()**, а также конструктор, который инициализирует все поля класса.

**Ссылка** на репозиторий **github.com** =

<https://github.com/themakson/MyJavaAppCourse2/tree/main/MyJavaAppCourse24>

**Вывод:**

В ходе лабораторной работы 4 мы доработали созданный ранее простой REST сервис с помощью фреймворка Spring Boot в языке программирования Java, клонировали сервис на 2 и протестировали их работу между собой, таким образом что отправив запрос будут взаимодействовать 2 сервиса одновременно, проверили их работу и логи.

|  |  |
| --- | --- |
| Сдал: | М.С. Ермаков |
| Проверил: | Н.А. Архипов |