**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 № 2

по дисциплине «Технологии Программирования»

Старший преподаватель: Н.А. Архипов

Студент гр. РИВ-220938у М.С. Ермаков

Екатеринбург 2023

**1. Цель работы:**

**Цель:** Разработать простой REST сервис

**Учебные вопросы:**

1) Описание входящего (request) и исходящего (response) сообщений, схема работы сервиса

2) Реализовать простой REST сервис

3) Реализовать дополнительный функционал в разработанном по примеру REST сервисе

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

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

**2. Описание задачи:**

**Описание входящего (request) и исходящего (response) сообщений, схема работы сервиса**

2.1 Описание входящего сообщения – request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Наименование ключа | Тип данных | Обязательность | Описание |
|  | uid | String | Да, до 32 символов | Уникальный идентификатор сообщение |
|  | operationUid | String | Да, до 32 символов | Уникальный идентификатор операции |
|  | systemName | String | Нет | Имя системы отправителя |
|  | systemTime | String | Да | Время создания сообщения |
|  | source | String | Нет | Наименование ресурса |
|  | communicationId | int | Минимум 1, максимум 100000 | Уникальный идентификатор коммуникации |
|  | templateId | int | Нет | Уникальный идентификатор шаблона |
|  | productCode | int | Нет | Код продукта |
|  | smsCode | int | Нет | Смс код |

Пример входящего сообщения:

{

"uid": "1",

"operationUid": "",

"systemName": "",

"systemTime": "",

"source": "",

"communicationId": 100,

"templateId": 8,

"productCode": 1500,

"smsCode": 10

}

1.2 Описание ответного сообщения – response

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Наименование ключа | Тип данных | Обязательность | Описание | Возможные значения |
|  | uid | String | Да | Уникальный идентификатор сообщение |  |
|  | operationUid | String | Да | Уникальный идентификатор операции |  |
|  | systemTime | String | Да | Имя системы отправителя | «ERP» |
|  | code | String | Да | Время создания сообщения | «success», «failed» |
|  | errorCode | String | Да | Наименование ресурса | «UnsupportedCodeException»,  «ValidationException»,  «UnknownException» |
|  | errorMessage | String | Да |  | «Не поддерживаемая ошибка», «Ошибка валидации», «Произошла непредвиденная ошибка» |

{

"uid": "1",

"operationUid": "",

"systemTime": "",

"code": "failed",

"errorCode": "ValidationException",

"errorMessage": "Ошибка валидации"

}

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. Чем отличаются GET запрос от POST запроса?

2. Какое содержимое может передаваться POST запросом?

3. Чем отличается XML файл от JSON?

4. Зачем нужны кастомные исключения?

5. Что означает аннотация @Service?

6. Что означает аннотация @RestController?

**3. Ход выполнения:**

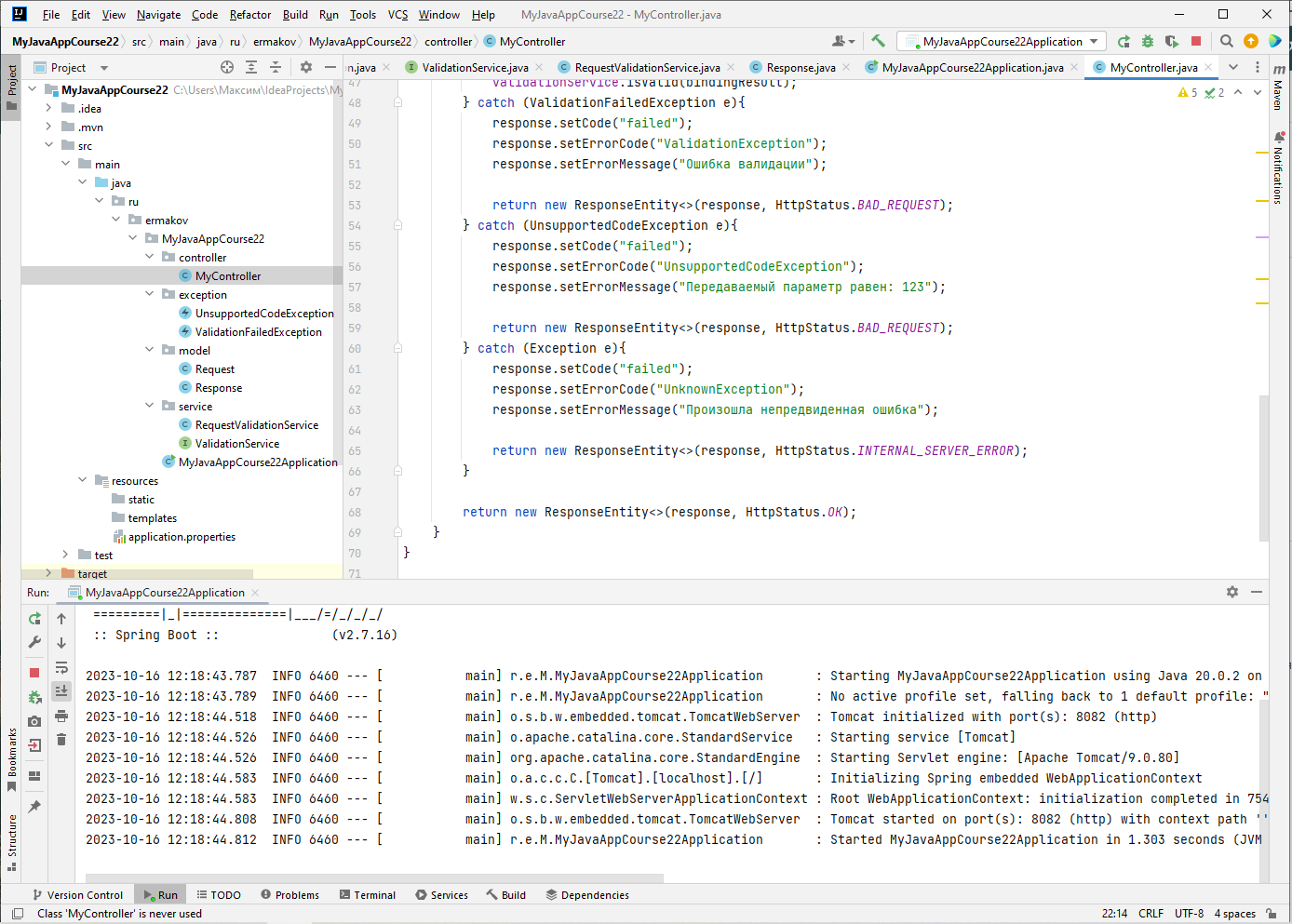


Рисунок 1 – Структура Проекта

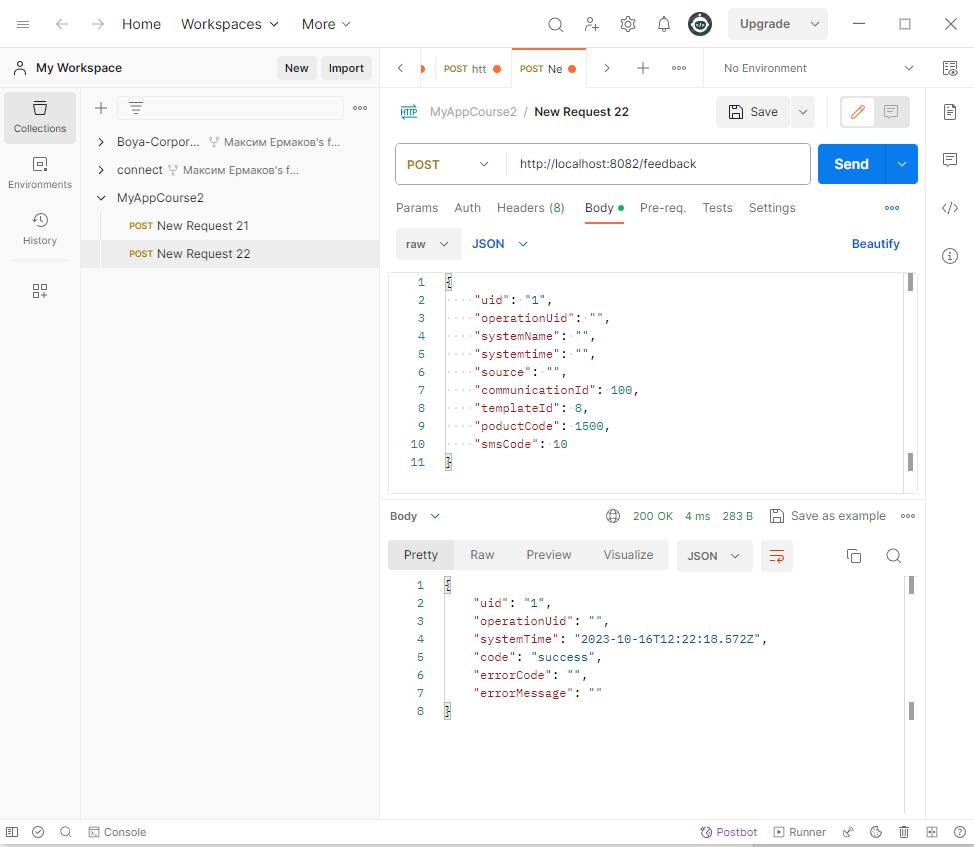


Рисунок 2 – Тестирование стандартного входящего сообщения через Post – Postman

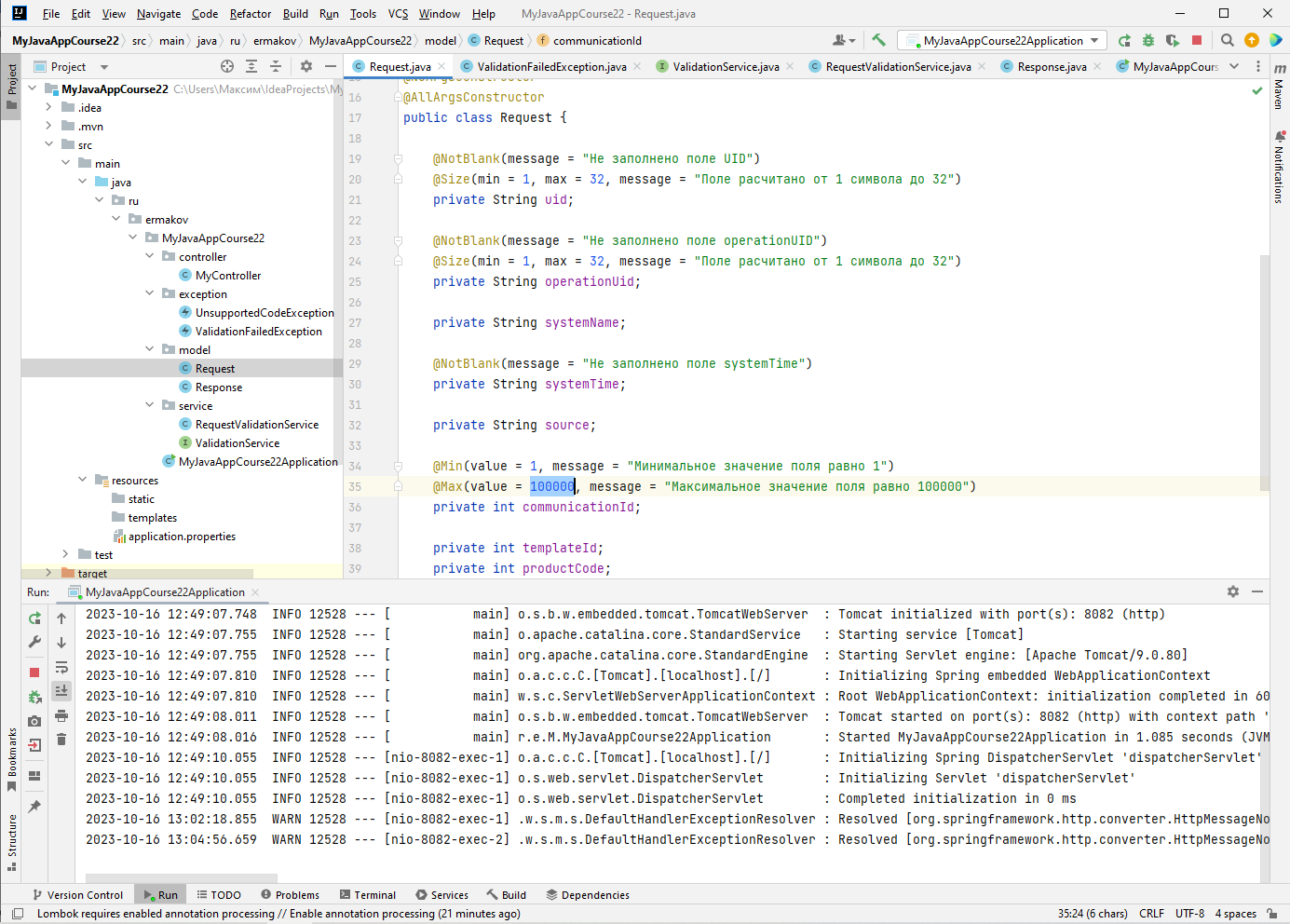


Рисунок 3 – Добавление валидации полей в соответствии с таблицей пункт 1.1

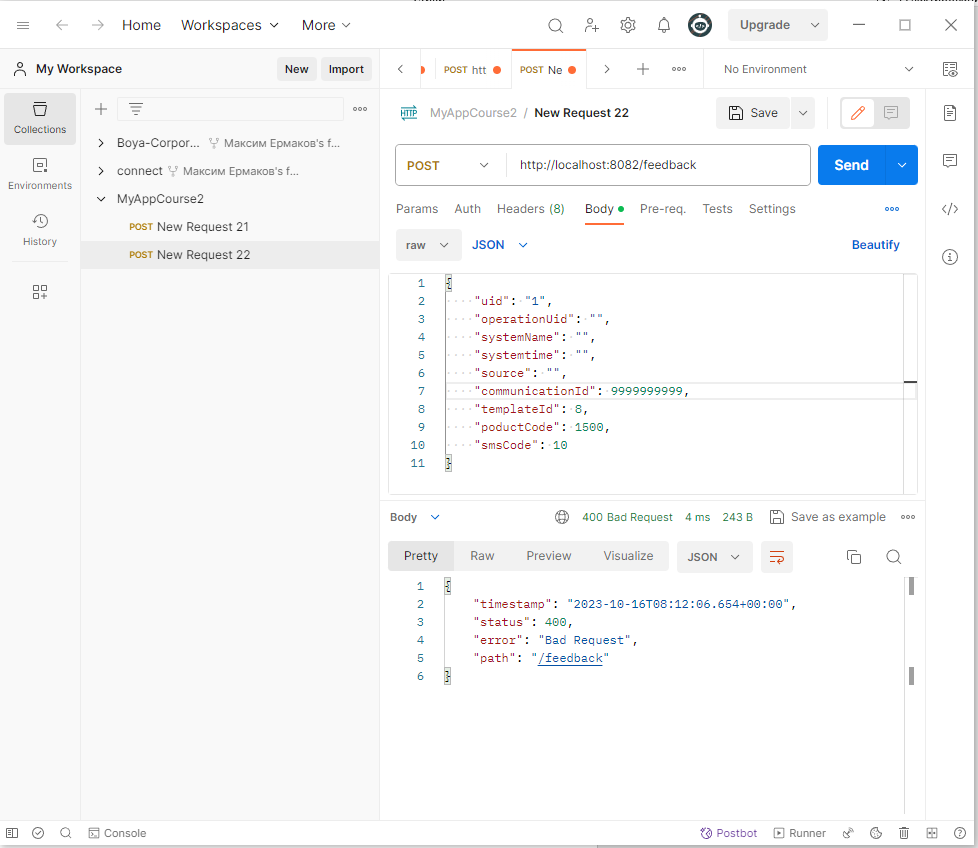


Рисунок 4 – Результат тестирования валидации с перегруженным полем communicationId

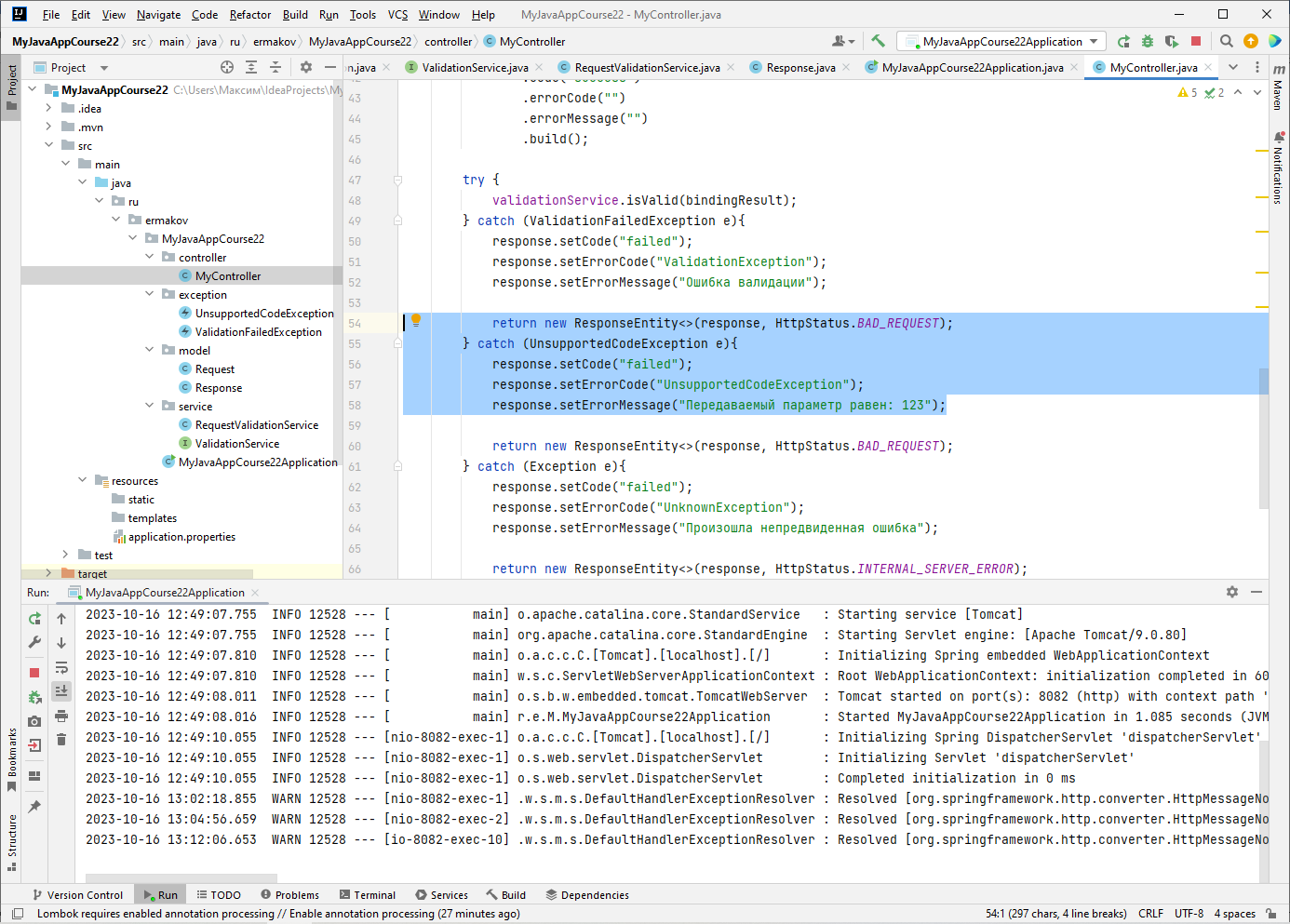


Рисунок 5 – Создание собственного исключения UnsopportedException в Mycontroller

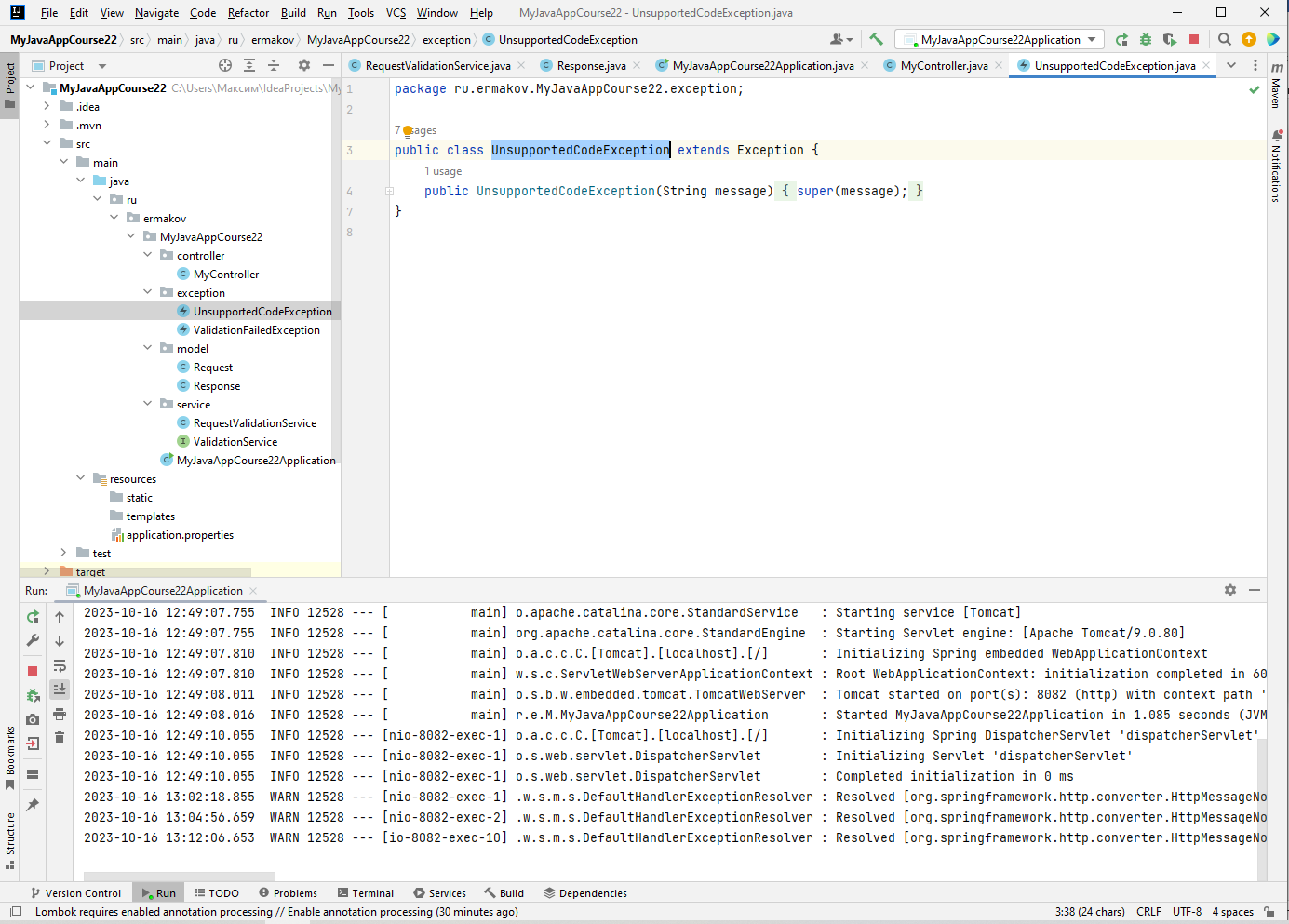


Рисунок 6 – Создание собственного исключения UnsopportedException в exception

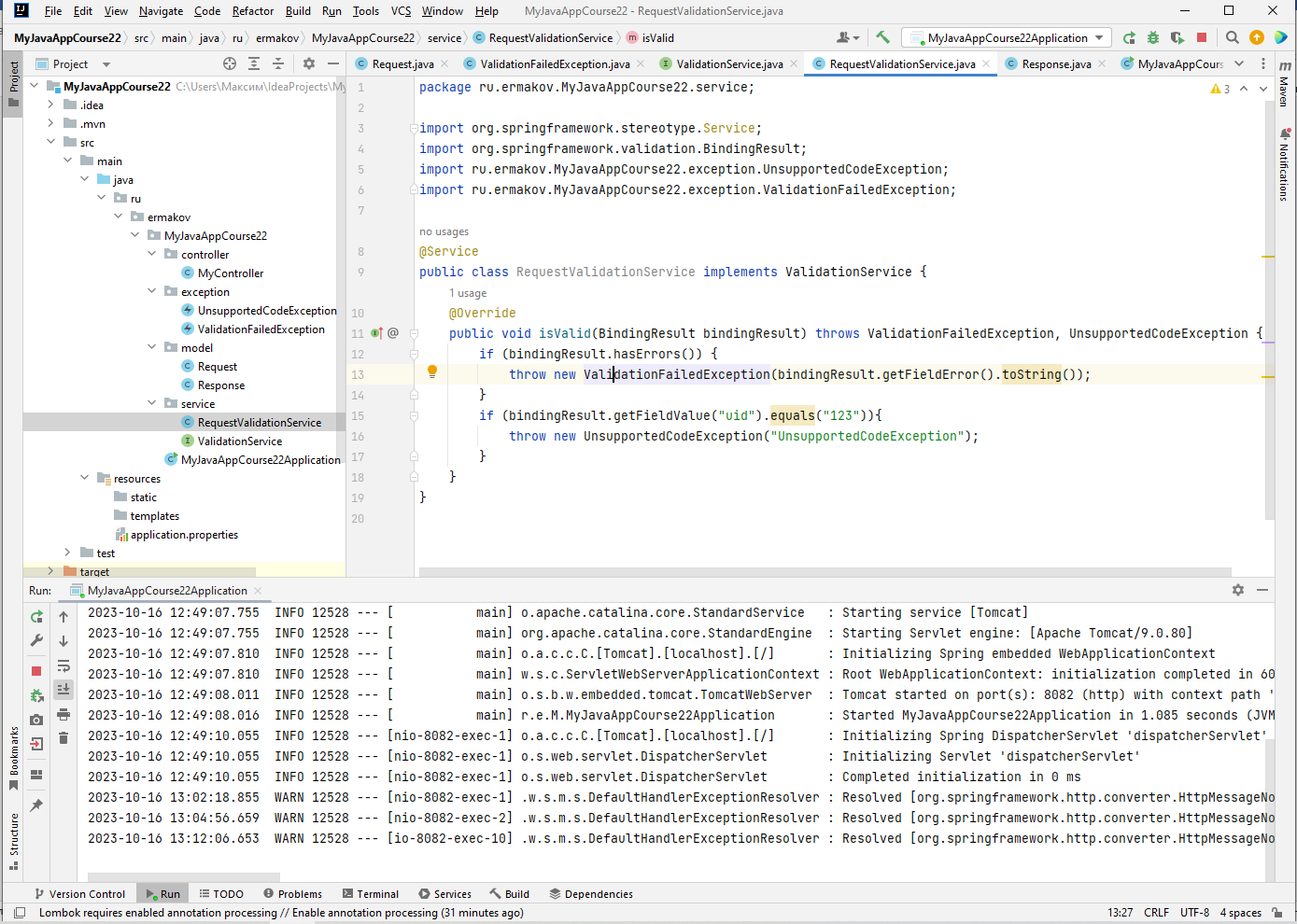


Рисунок 7 – Создание собственного исключения UnsopportedException в service

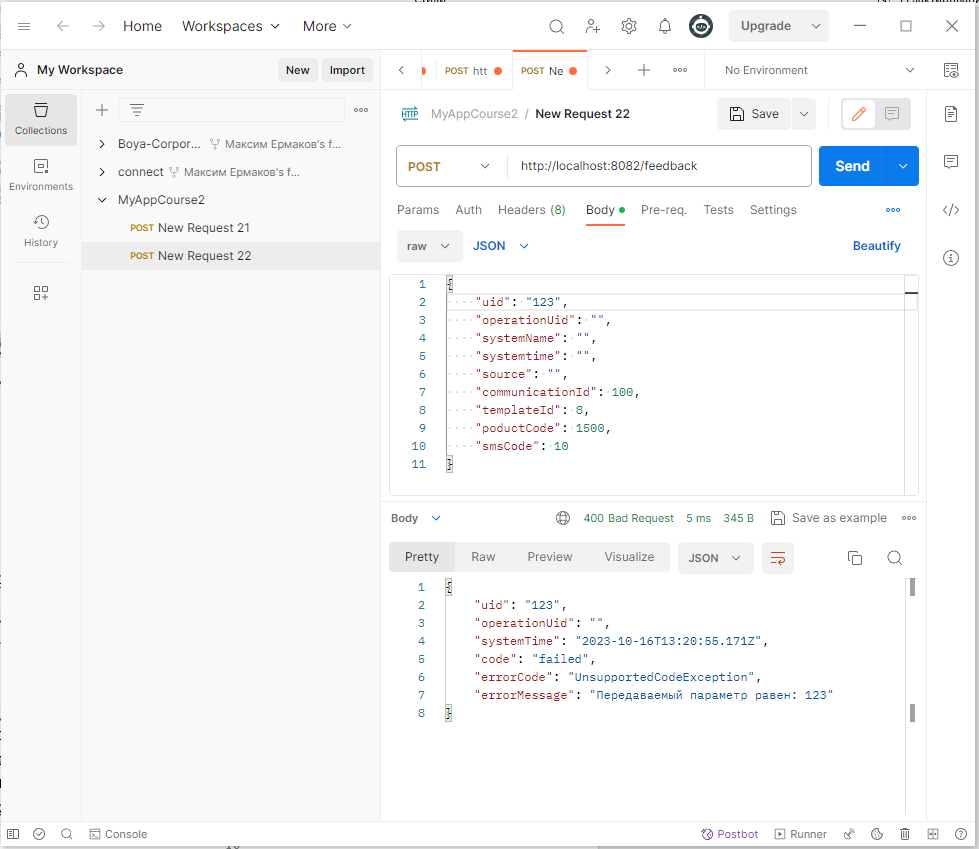


Рисунок 8 – Тестирование собственного исключения UnsopportedException через POST – Postman

**Ответы на вопросы:**

2) Какое содержимое может передаваться POST запросом?

В программировании POST — один из многих методов запроса, в целях поддерживаемых HTTP протоколом, используемым во Всемирной паутине. **Метод запроса POST предназначен для направления запроса, при котором веб-сервер принимает данные, заключённые в тело сообщения, для хранения. Он часто используется для загрузки файла или представления заполненной веб-формы.**

4) Зачем нужны кастомные исключения?

Для определенных ситуаций в своей программе нужны кастомные исключения. Это может быть полезно в случаях, когда встроенных исключений недостаточно для обработки определенных ситуаций в нашей программе. Например мы можем создать исключения для возраста пользователя, В данном случае, если возраст пользователя меньше 18 лет, мы выбрасываем наше кастомное исключение с сообщением об ошибке. Если возраст пользователя меньше 18 лет, программа выведет сообщение.

6) Что означает аннотация @RestController?

**Аннотация @RestController**

Аннотация @RestController появилась в Spring 4.0 и объединяет в себе аннотации @Controller и @ResponseBody. Это означает, что она не только помечает класс как Spring MVC Controller, но и автоматически преобразует возвращаемые контроллером данные в формат JSON или XML.

Таким образом, основное различие между @Controller и @RestController заключается в том, что @Controller используется для обработки веб-страниц, а @RestController — для создания RESTful веб-сервисов, которые возвращают данные в формате JSON или XML.

Аннотация @Controller может быть использована и для Web MVC, и для REST приложений, но в случае REST приложений к каждому методу контроллера придется добавлять аннотацию @ResponseBody для автоматического преобразования ответа в формат JSON или XML. Определить, MVC это приложение или REST, можно по наличию или отсутствию аннотации @ResponseBody в методах контроллера.

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

<https://github.com/themakson/MyJavaAppCourse2/tree/main/MyJavaAppCourse22>

**Вывод:**

В ходе лабораторной работы 2 мы научились создавать простой REST сервис с помощью фреймворка Spring Boot в языке программирования Java, познакомились с валидацией полей, протестировали данную валидацию через POST запрос в приложении POSTMAN, а также создали собственное исключение UnsupportedException, и также протестировали его.

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