**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 № 3

по дисциплине «Программирование»

Старший преподаватель: Н.А. Архипов

Студент гр. РИЗ-220938у А.А. Фищев

Екатеринбург 2023

2. Цель работы: Доработать простой REST сервис.

3. Описание задачи:

1. 1. Доработать логику изменения исходящего response
   2. Реализовать простой REST сервис
   3. Добавить и настроить логирование в приложении
   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. Ход выполнения:
   1. Доработанная логика исходящего response

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

* 1. Доработан REST сервис, настроено и добавлено логирование нескольких уровней.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

* 1. Ответы на вопросы
     1. Что такое CRM система? Приведите примеры. CRM (Customer Relationship Management) — это программное обеспечение для управления взаимоотношениями с клиентами. Она позволяет компаниям собирать, хранить и анализировать информацию о своих клиентах, а также управлять этими данными для улучшения качества обслуживания и увеличения продаж.

Примеры CRM систем включают Salesforce, Documentum, Microsoft Dynamics 365 и Oracle CRM. Каждая из этих систем имеет свои особенности и функциональность, но все они направлены на упрощение работы с клиентами и повышение эффективности бизнеса. Чем отличается XML файл от JSON? В целом, выбор между XML и JSON зависит от требований проекта и предпочтений разработчика. Если требуется строгое форматирование данных и использование определенных типов элементов, то лучше использовать XML. Если же важна гибкость и возможность работы с различными типами данных, то лучше выбрать JSON.

* + 1. Какие библиотеки логгирования бывают в Spring? В Spring есть несколько библиотек логгирования, которые позволяют записывать логи в файлы или базы данных. Некоторые из них:

1. Logback — это библиотека логгирования, которая использует механизм аннотаций для записи логов в файловые потоки. Она поддерживает различные форматы файлов, такие как JSON, XML и текстовый формат.

2. Log4j — это еще одна популярная библиотека логгирования, которая использует Java-контейнер Logback для записи логов в файловые потоки. Она поддерживает различные форматы файлов, включая JSON, XML и текстовый формат.

3. org.slf4j — это библиотека логгирования, которая использует механизм аннотаций для записи логов в файловые потоки. Она поддерживает различные форматы файлов, включая JSON, XML и текстовый формат.

4. java.util.logging — это стандартная библиотека логгирования, которая предоставляет доступ к различным типам логгирования, таким как потоковое чтение, запись в файлы и базы данных. Она может быть использована вместе с другими библиотеками логгирования для создания более сложных систем логгирования.

* + 1. Что означает аннотация @Qualifier? Аннотация `@Qualifier` используется в Spring Framework для указания имени класса, который будет использоваться в качестве значения аннотации. Это позволяет указать имя класса, которое должно быть использовано вместо фактического имени класса при вызове метода.

Например, если у вас есть класс «Person», который содержит поле «name», то вы можете использовать аннотацию «@Qualifier("person")», чтобы указать, что метод «getName()» должен быть вызван вместо метода «getName()» класса «Person». Это позволит избежать конфликтов имен методов и сделает код более читаемым и поддерживаемым.

Кроме того, аннотация «@Qualifier» может использоваться для определения пространства имен, которое будет использоваться вместо фактического пространства имен при вызове метода. Например, если у вас есть класс «MyClass», который содержит поле «myField», то вы можете использовать аннотацию «@Qualifier("myNamespace")», чтобы указать, что метод «getMyField()» должен быть вызван вместо метода «getMyField()» класса «MyClass». Это позволит избежать конфликтов имен методов и сделает код более читаемым и поддерживаемым

* 1. Так же прикрепляю ссылку на [GIT РЕПОЗИТОРИЙ](https://github.com/papakarlorabotaet/lrForStudySecondYear)

5. Вывод: за время выполнения лабораторной работы укрепил навыки программирования и доработал простой REST сервис