**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 № 6

по дисциплине «Программирование»

Старший преподаватель: Н.А. Архипов

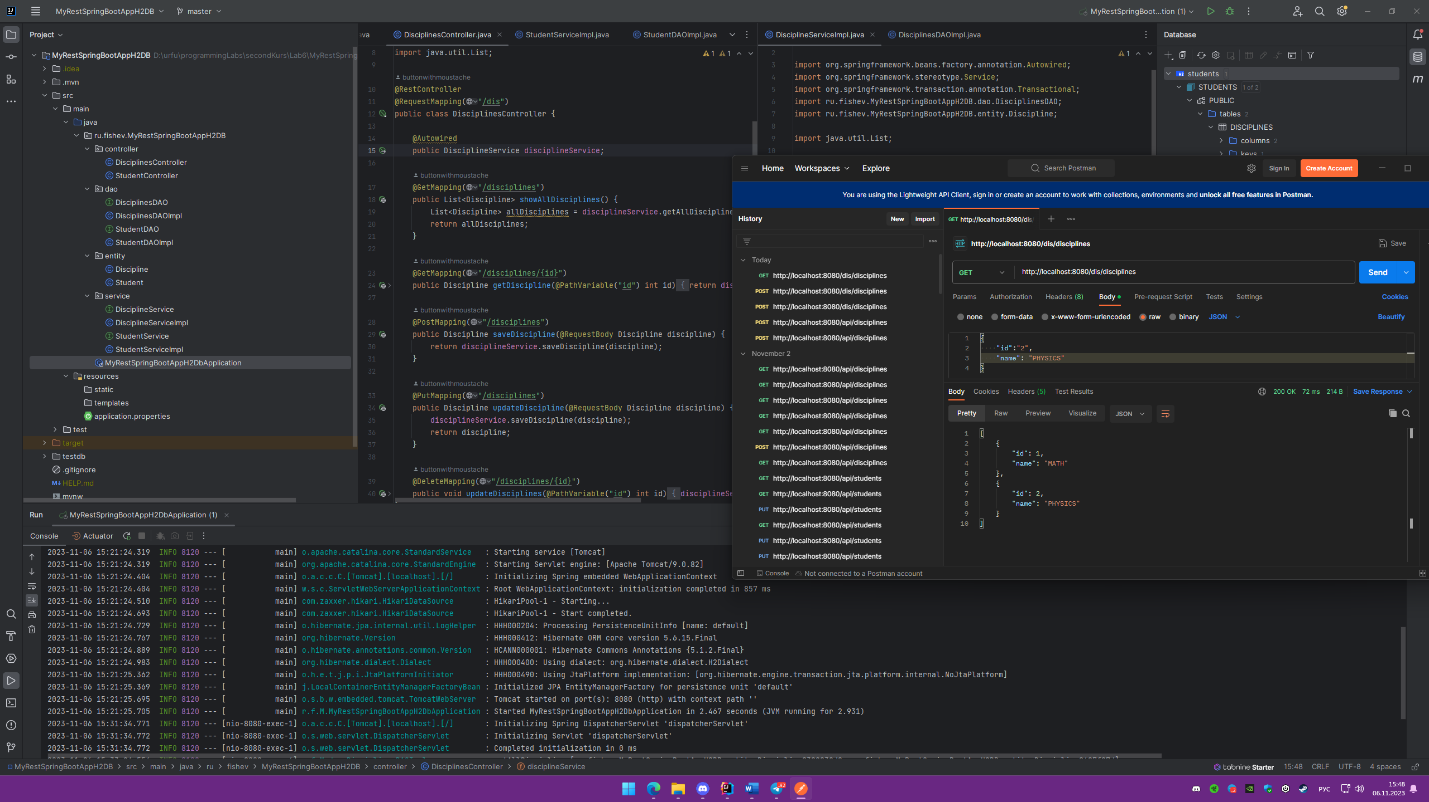
Студент гр. РИЗ-220938у А.А. Фищев

Екатеринбург 2023

2. Цель работы: разработать REST-full-service.

3. Описание задачи:

1. 1. Доработайте методы, возвращающие значение, чтобы возвращался ответ об успехе или неуспехе завершения операции.
   2. Создайте вторую таблицу в базе данных «Учебный дисциплины» и опишите для нее необходимый контроллер, сущность, DAO и сервисный слой.
   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
2. Ход выполнения:
   1. Ответ постмана от новой таблице в базе данных



* 1. Ответы на вопросы
     1. Что из себя представляют подходы API-First и Code-First? Подход API-First (Application Programming Interface First) предполагает разработку программного интерфейса (API) для приложения до его создания. Это означает, что разработчики сначала определяют структуру и функциональность API, а затем используют его для создания приложения.

Подход Code-First (Code First) подразумевает создание кода приложения перед созданием API. В этом случае разработчики сначала создают функциональность приложения, используя языки программирования и фреймворки, а затем используют эту функциональность для создания API.

Оба подхода имеют свои преимущества и недостатки. Подход API-First может быть полезен для разработки больших и сложных приложений, где необходимо четкое определение структуры и функциональности API. Подход Code-First может быть более гибким и быстрым в разработке небольших приложений. Что такое интеграционное тестирование? Интеграционное тестирование — это процесс проверки взаимодействия между различными компонентами программного обеспечения. Оно заключается в проверке корректности работы системы при объединении нескольких компонентов в единое целое. В ходе интеграционного тестирования проверяется работа системы как единого целого, а не отдельных ее частей. Это помогает выявить проблемы, связанные с неправильной работой компонентов, их взаимодействием и передачей данных между ними.

* + 1. Какие существуют виды итеграция инфомационных систем?

Существует несколько видов интеграции информационных систем:

* + - 1. Локальная интеграция — это процесс объединения данных между двумя или более системами на локальном уровне. Например, объединение баз данных или подключение к другим приложениям через сетевой протокол.
      2. Глобальная интеграция — это процесс объединения данных между различными географическими регионами или странами. Например, интеграция баз данных в разных странах для обмена информацией о клиентах и заказах.
      3. Интеграция на основе стандартов — это процесс использования стандартных протоколов и форматов данных для обмена информацией между системами. Например, использование стандарта ODBC для доступа к данным в различных базах данных.
      4. Интеграция на основе API — это процесс использования API (Application Programming Interface) для обмена информацией между системами. Например, использование API Google Maps для получения информации о местоположении пользователя.
      5. Интеграция на основе технологий — это процесс использования технологий, таких как веб-сервисы, для обмена информацией между системами. Например, использование веб-сервисов для отправки сообщений между приложениями
    1. Что такое gRPC, и какая разница между ним и REST?

gRPC (Google Remote Procedure Call) — это протокол удаленных вызовов процедур, который позволяет обмениваться данными между клиентом и сервером. Он основан на протоколе HTTP/2 и использует JSON-подобные сообщения для передачи данных.

REST (Representational State Transfer) — это архитектурный стиль, который определяет набор правил для обмена данными между клиентом и сервером. REST использует HTTP-протокол и формат данных JSON или XML для передачи данных.

Основная разница между gRPC и REST заключается в том, что gRPC является более низкоуровневым протоколом, чем REST. gRPC предоставляет более прямой доступ к ресурсам сервера и может обрабатывать большие объемы данных быстрее, чем REST. Однако gRPC требует больше ресурсов и времени для разработки, чем REST.

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

5. Вывод: за время выполнения лабораторной работы укрепил навыки программирования и разработал REST-full-service.