Министерство образования и науки Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования  
«Уральский федеральный университет имени первого Президента Рос- сии Б. Н. Ельцина»  
Институт радиоэлектроники и информационных технологий – РтФ

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

ПО ДИСЦИПЛИНЕ «Программирование на Java»

Преподаватель: Архипов Н. А.

Студент: Мишунин А.И.

Группа: РИМ-220971

Екатеринбург 2023

ЛАБОРАТОРНАЯ РАБОТА №1

**Цель:**

Создание простейшего веб-приложения с помощью фреймворка Spring Boot

**Необходимое программное обеспечение:**

IntelliJ IDEA CE, Spring Boot 2, Java 8+, Browser

**Полезные ссылки:**

1) <https://start.spring.io/>

2) <https://spring.io/quickstart>

**Задачи:**

1) Реализовать простейшее Spring Boot приложение

2) Реализовать дополнительный функционал в разработанном по примеру Spring Boot приложении

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

Содержание:

[1 Реализация простейшего Spring Boot Application 2](#_Toc144894801)

[2 Реализация дополнительного функционала 10](#_Toc144894802)

[3 Вопросы 11](#_Toc144894803)

[4 Требования к оформлению отчета 12](#_Toc144894804)

# **1 Реализация простейшего Spring Boot Application**

1. Зашли на сайт <https://start.spring.io/>, создали конфигурацию. Открыли файл.

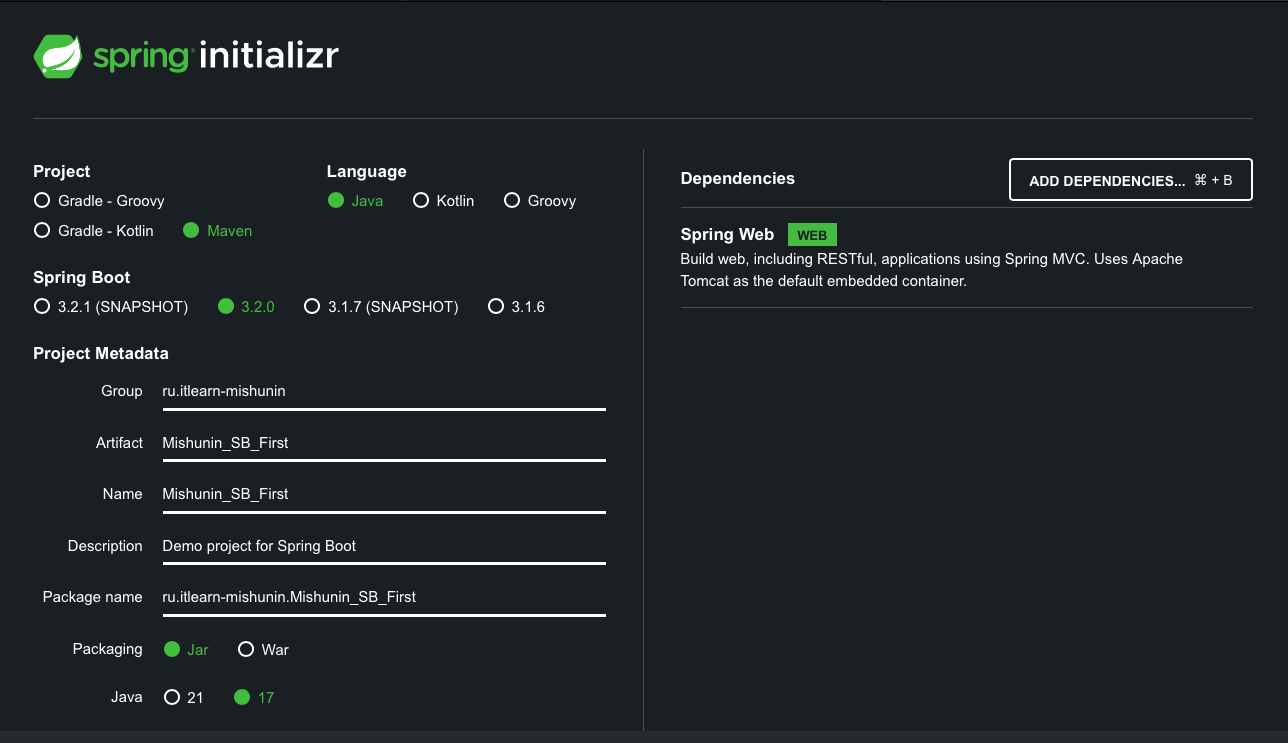


Рисунок 1 – Генерация конфигурации приложения

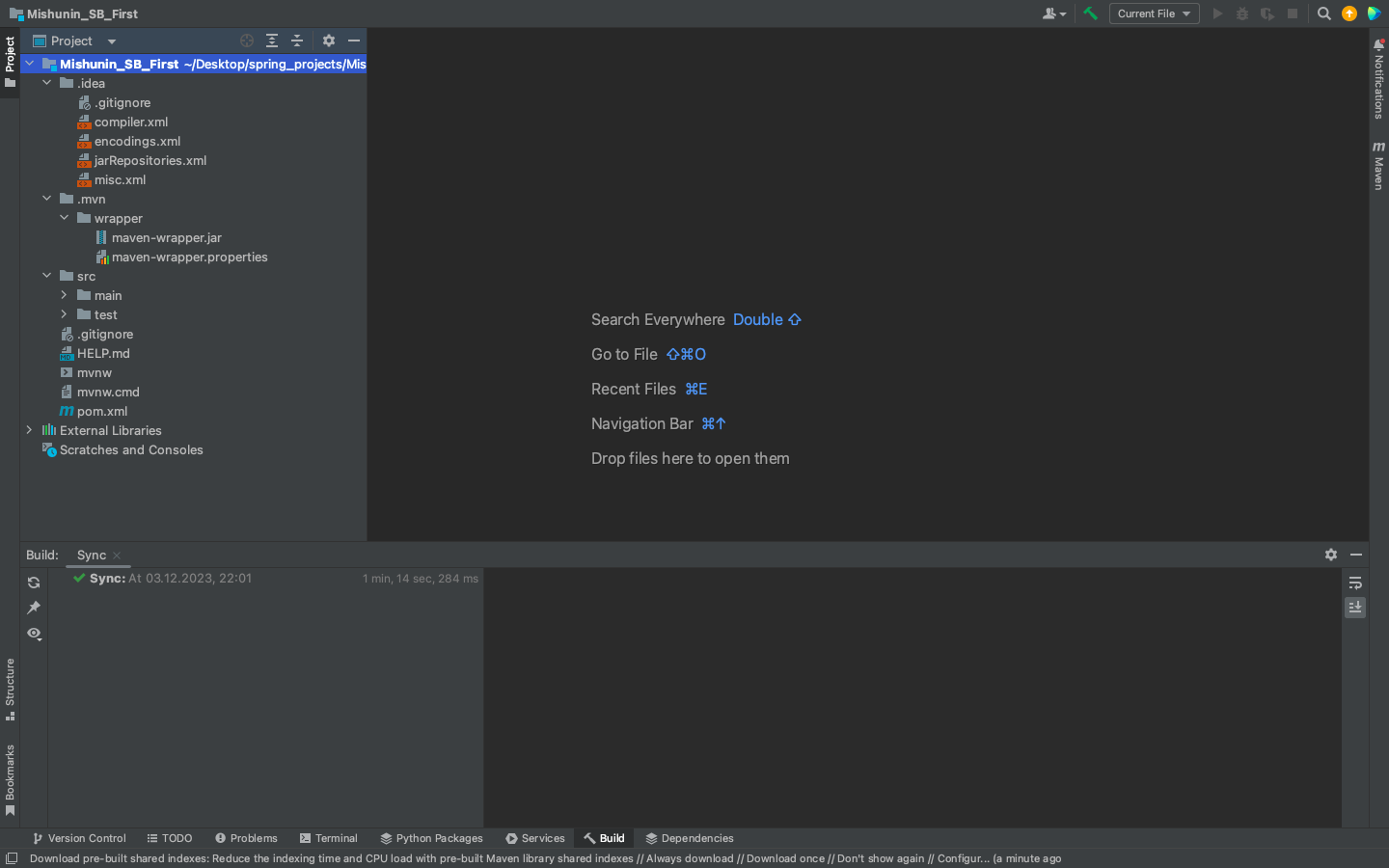


Рисунок 2 – Окно IDEA после открытия и загрузки зависимостей

3. Создали в проекте пакет с названием **hello,** в этом пакете создали класс, назвали его **HelloController,** в нем описали метод hello().

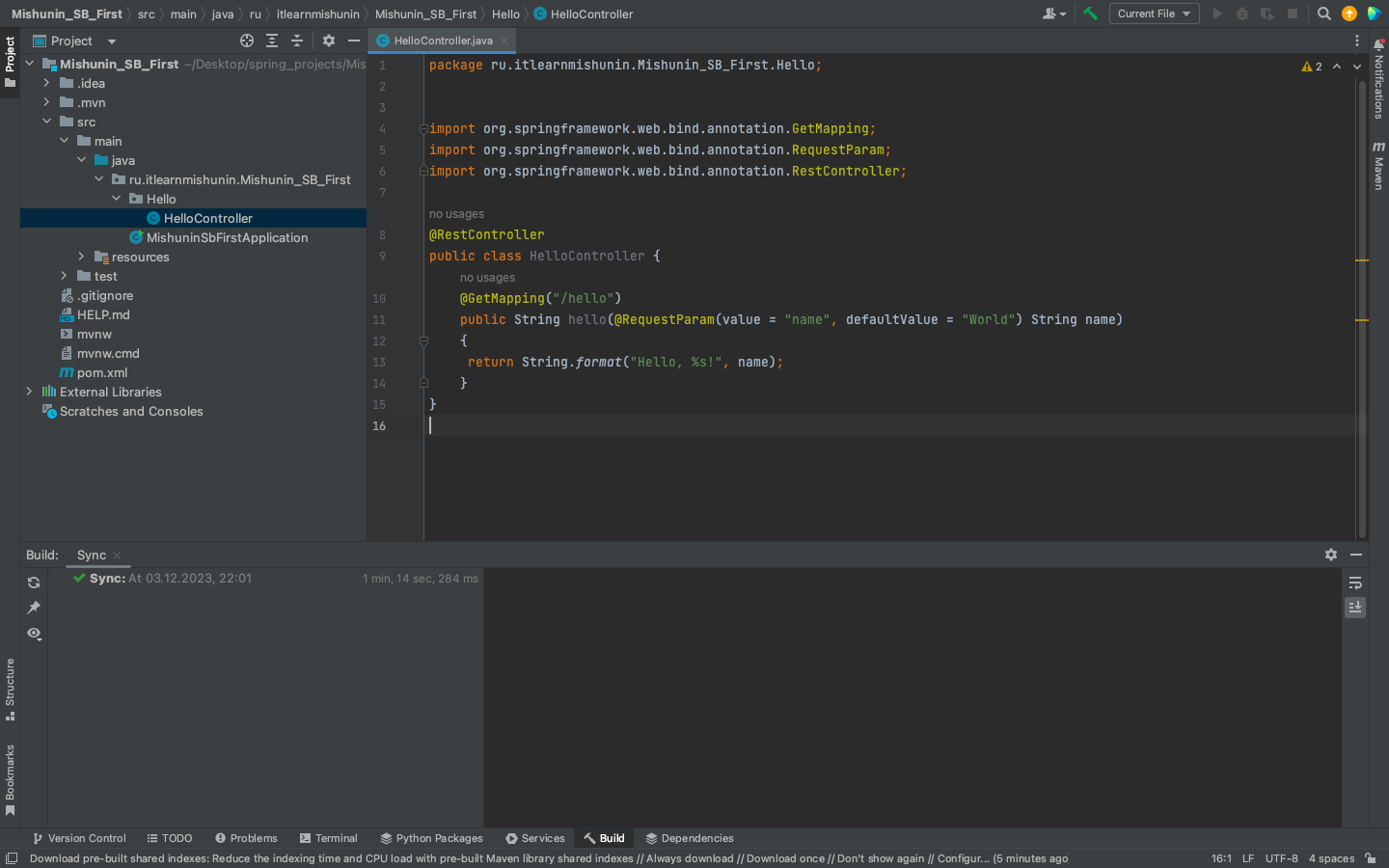


Рисунок 3 – Создание класса HelloController и метода hello

4. Перешли в класс **MishuninSbFirstApplication** и запустили приложение:

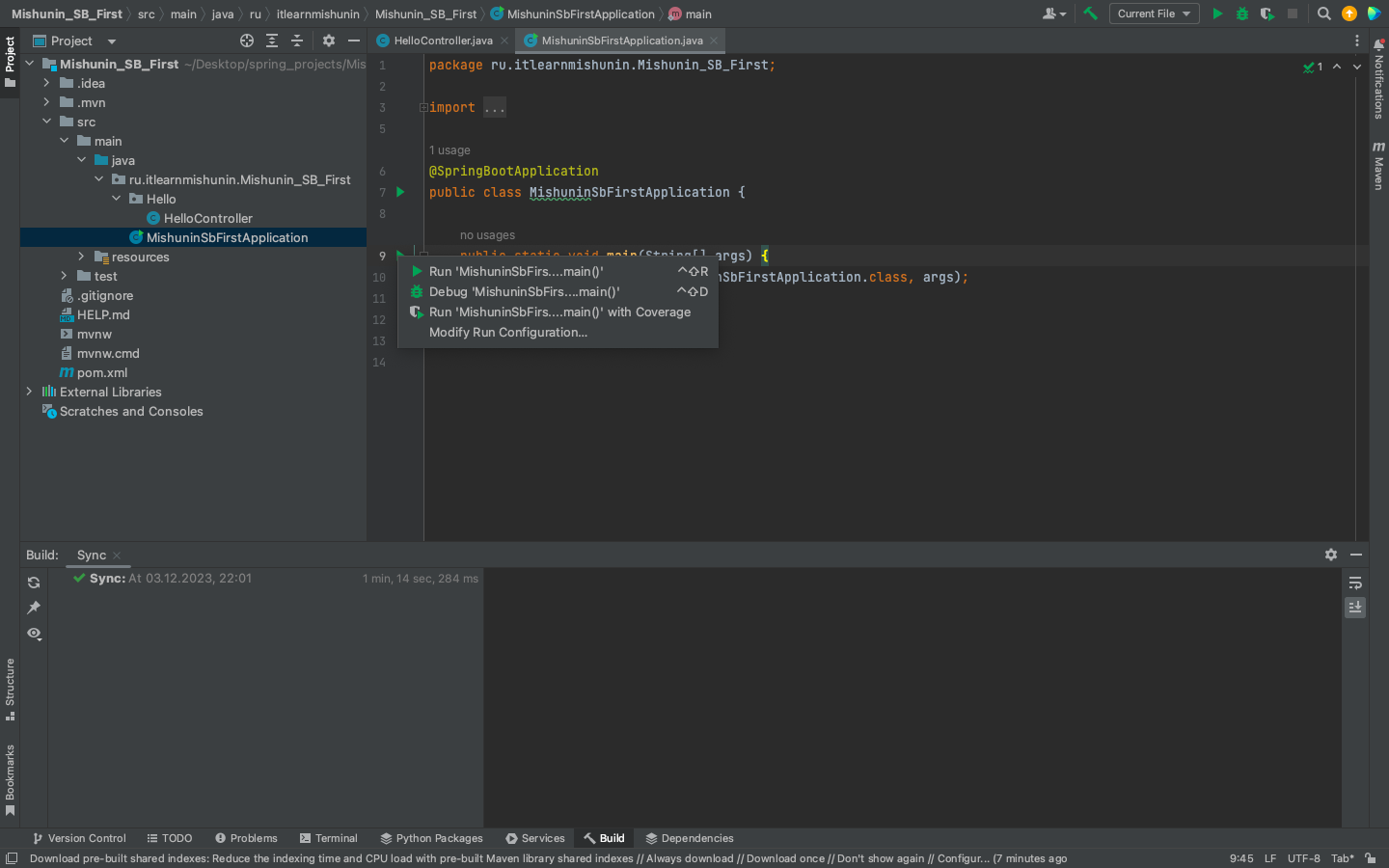


Рисунок 4 – Запуск приложения

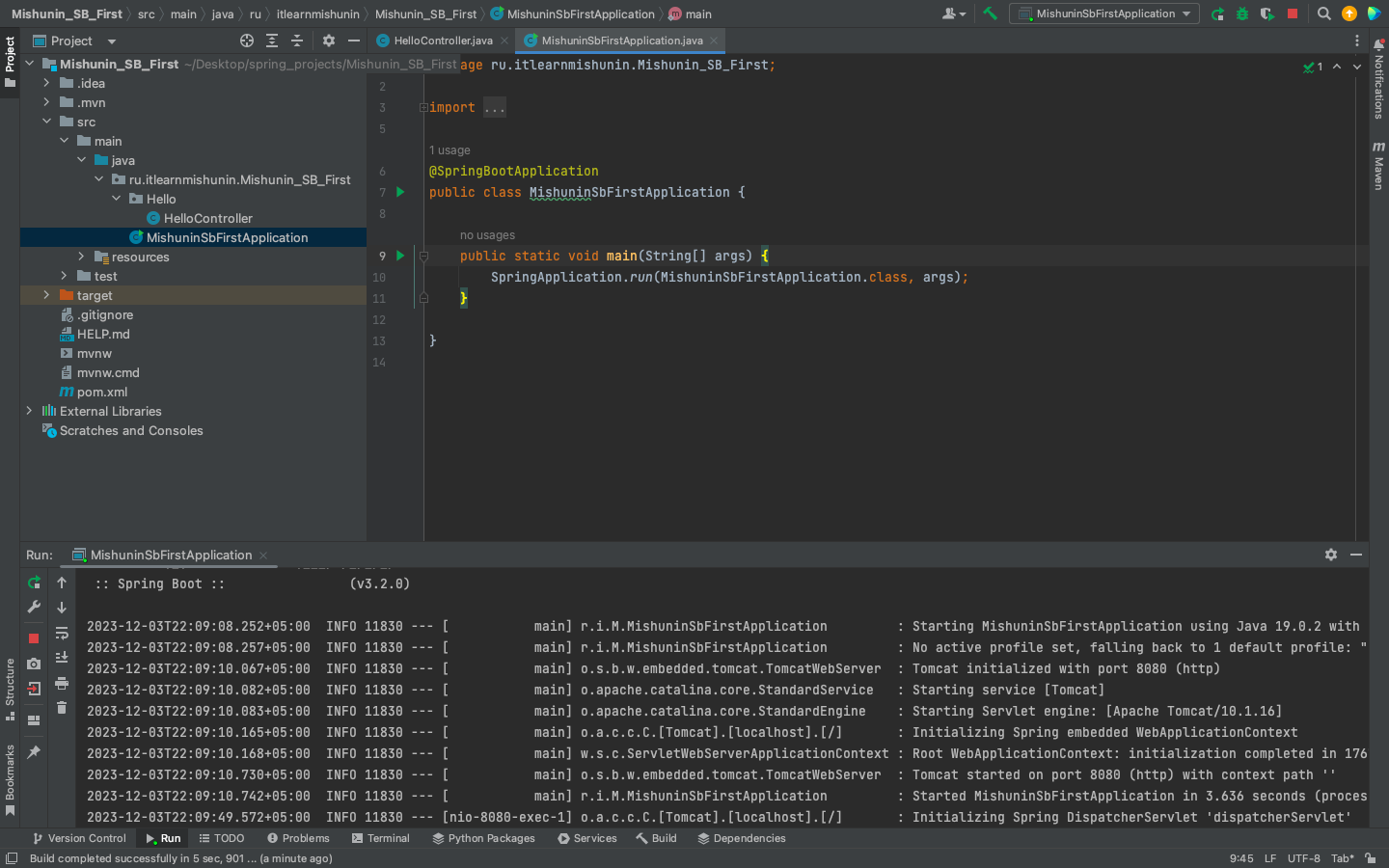


Рисунок 5 – Старт приложения и вывод логов в консоль

6. Протестировали приложение, зайдя в браузер и в строке поиска введя [**http://localhost:8080/hello**](http://localhost:8080/hello),открыли страницу с вернувшейся строкой указанной в методе hello()

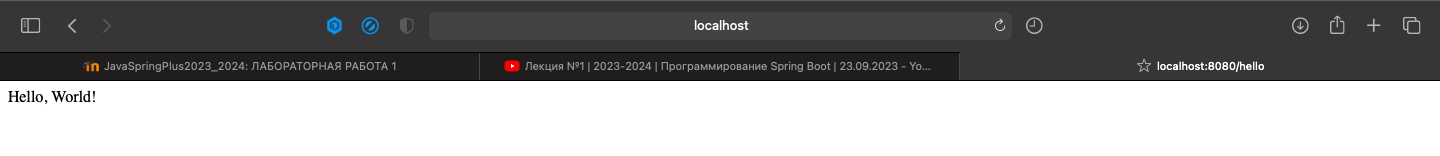


Рисунок 6 – Рузультат тестирования запроса [**http://localhost:8080/hello**](http://localhost:8080/hello)

7. Протестировали приложение, зайдя в браузер и в строке поиска введите http://localhost:8080/hello?name="Cool\_boy3000",открылась страница с вернувшейся строкой указанной в методе hello(), выполнилась логика добавления параметра переданного в метод GET

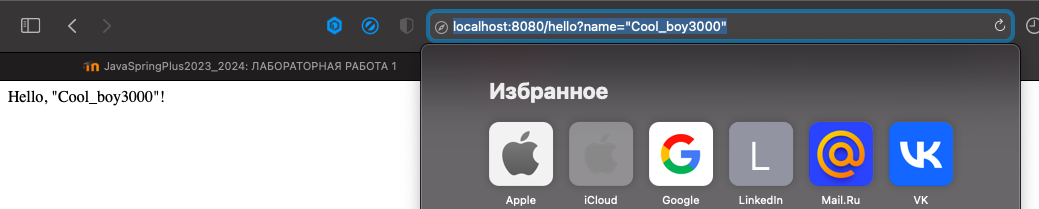
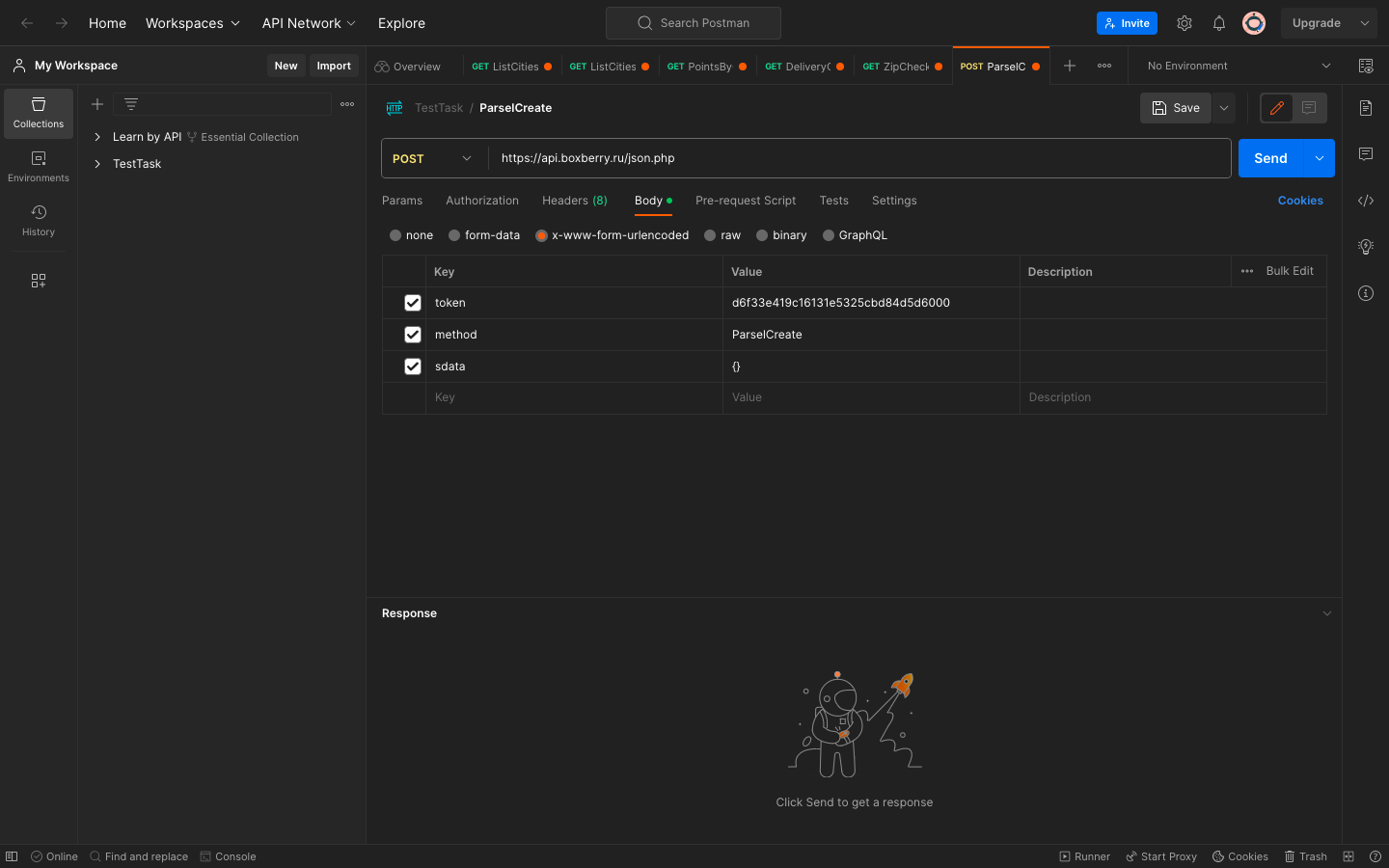


Рисунок 7 – Результат тестирования запроса http://localhost:8080/hello?name="Cool\_boy3000"

8. Установили программу Postman и протестировали работу приложения с помощью Postman



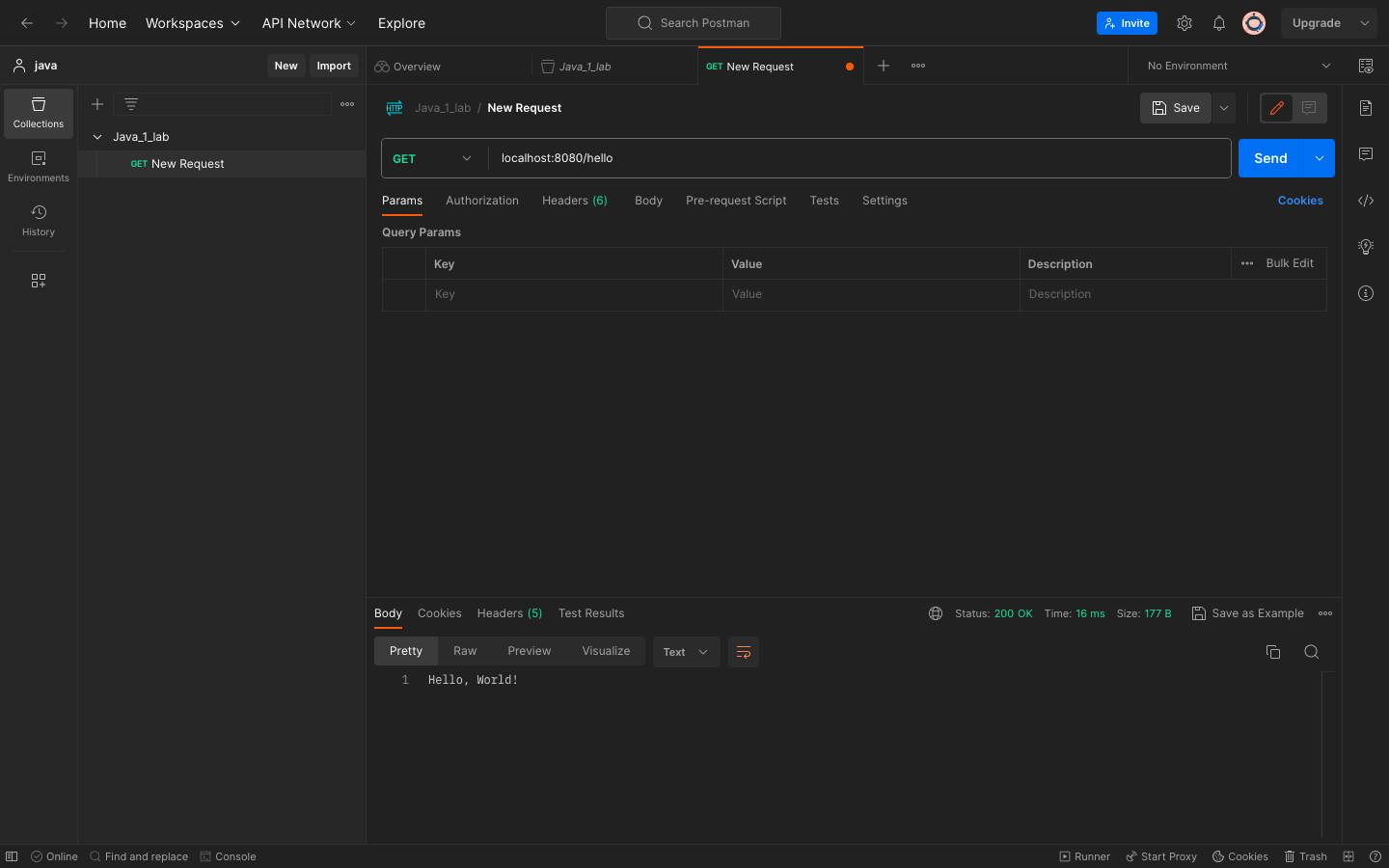


Рисунок 8 – Результат тестирования запроса [**http://localhost:8080/hello**](http://localhost:8080/hello)с помощью Postman

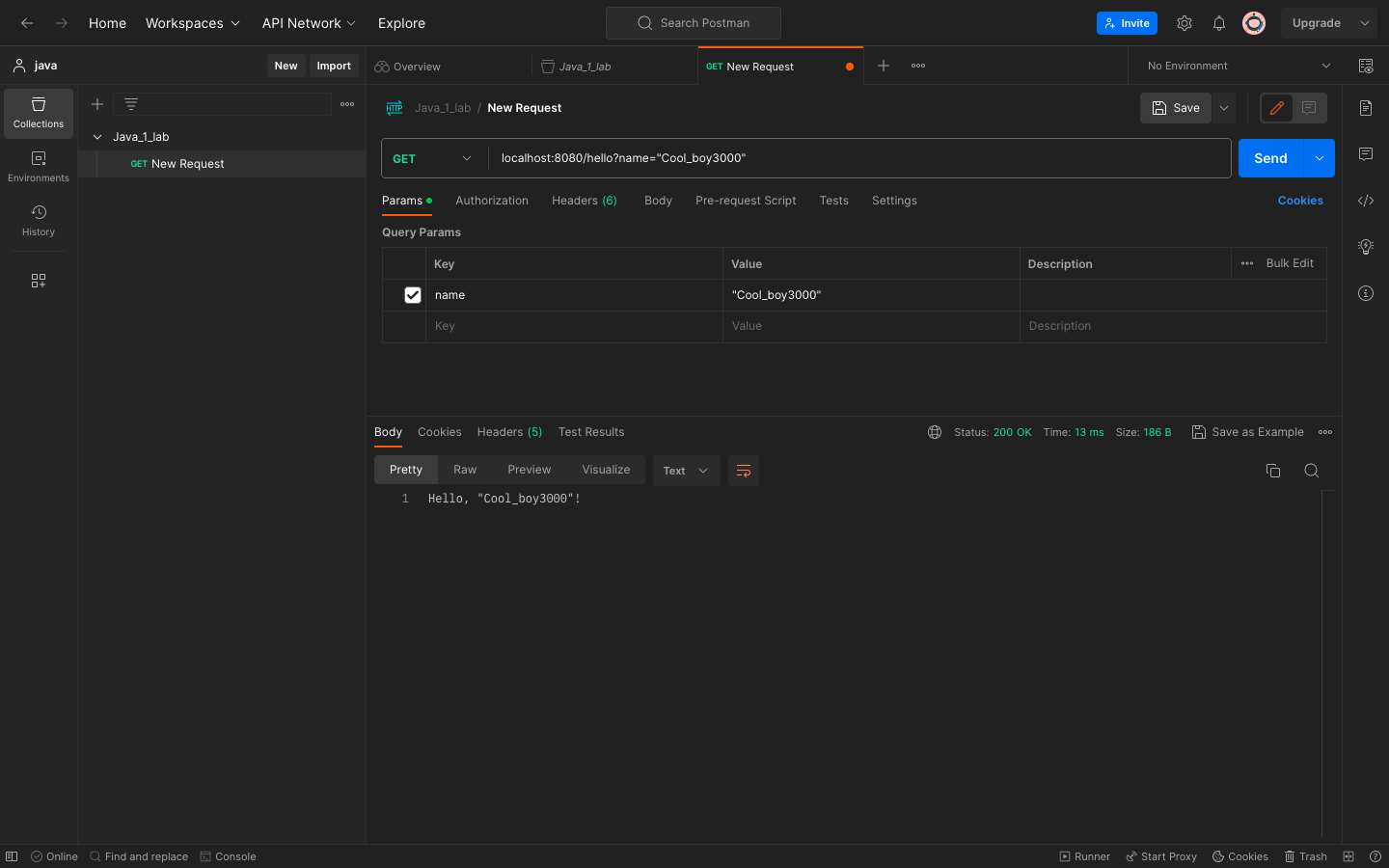


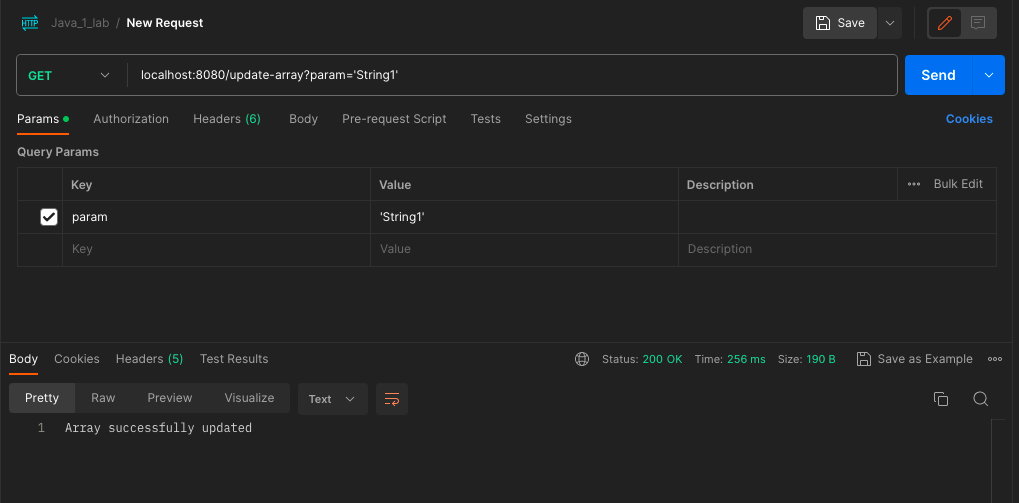
Рисунок 9 – Результат тестирования запроса [http://localhost:8080/hello?name="Cool\_boy3000](http://localhost:8080/hello?name=%22Cool_boy3000)" с помощью Postman

# 2 Реализация дополнительного функционала

1. Создайте GET-метод **updateArrayList**(String s), который по url “/update-array” принимает аргумент и записывает его в ArrayList<String >, реализуйте логику - если это первый вызов метода, то создается пустой ArrayList, если он не пустой, то туда записывается значение.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

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

Автоматически созданное описание

2. Создайте GET метод **showArrayList()**, который по url “/show-array” возвращается все элементы хранящиеся в ArrayList <String >, созданном в п. 1

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

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

Автоматически созданное описание

3. Создайте GET-метод **updateHashMap**(String s), который по url “/update-map” принимает аргумент и записывает его в HashMap< Integer, String>, реализуйте логику - если это первый вызов метода, то создается пустой HashMap, если он не пустой, то туда записывается значение.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

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

Автоматически созданное описание

Изображение выглядит как программное обеспечение, Мультимедийное программное обеспечение, текст, Графическое программное обеспечение

Автоматически созданное описание

4. Создайте GET метод **showHashMap ()**, который по url “/show-map” возвращается все элементы хранящиеся в HashMap<Integer, String>, созданном в п. 3

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

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

Автоматически созданное описание

5. Создайте GET метод **showAllLenght ()**, который по url “/show-all-lenght” возвращает текст, в котором указано количество элементов в ArrayList и HashMap

Изображение выглядит как снимок экрана, текст, Мультимедийное программное обеспечение

Автоматически созданное описание

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

Автоматически созданное описание

2. Чем отличаются языки программирования Java и Kotlin?

1. Синтаксис: Java использует фигурные скобки для определения блоков кода, тогда как Kotlin использует круглые скобки. Кроме того, в Kotlin можно использовать отступы вместо точек с запятой для обозначения блоков кода.

2. Поддержка функционального программирования: Kotlin полностью поддерживает функциональное программирование и имеет встроенные возможности для работы с неизменяемыми данными (immutable data), что делает его более гибким языком программирования.

3. Наследование: В Java наследование может быть использовано только для создания иерархий классов, тогда как в Kotlin наследование может использоваться также для создания объектов.

4. Управление памятью: Java является языком со сборщиком мусора, который автоматически освобождает память от неиспользуемых объектов. В то время как Kotlin использует ленивую семантику, которая откладывает освобождение памяти до тех пор, пока объект не будет удален из памяти.

5. Компиляция: Java компилируется в байт-код, который затем выполняется на виртуальной машине Java (JVM). Kotlin компилируется напрямую в байт-код JVM, что делает его быстрее и эффективнее.

В целом, Java и Kotlin очень похожи друг на друга, но имеют некоторые отличия в синтаксисе и возможностях. Выбор между ними зависит от конкретного проекта и потребностей разработчика.

4. Чем отличаются языки программирования Kotlin и Groovy?

1. Синтаксис: Kotlin и Groovy используют разные синтаксисы. Kotlin использует стандартный синтаксис Java-подобных языков, а Groovy имеет свой собственный синтаксис, который более удобен для написания скриптов.

2. Поддержка функционального программирования: Оба языка поддерживают функциональное программирование, но Kotlin обеспечивает более полную поддержку этого подхода благодаря своей конструкции неизменяемых данных (immutable data).

3. Наследование: Оба языка поддерживают наследование, но Groovy имеет более гибкую систему наследования, которая позволяет использовать наследование для создания объектов.

4. Управление памятью: Оба языка используют сборщик мусора для автоматического освобождения памяти от неиспользуемых объектов. Однако Groovy имеет более эффективную систему управления памятью благодаря своей конструкции неизменяемых данных (immutable data).

5. Компиляция: Оба языка компилируются в байт-код JVM, который затем выполняется на виртуальной машине Java (JVM). Однако Groovy может быть скомпилирован в байт-код JVM с помощью инструментария Gradle, тогда как Kotlin может быть скомпилирован напрямую в байт-код JVM.

В целом, Kotlin и Groovy очень похожи друг на друга, но имеют некоторые отличия в синтаксисе и возможностях. Выбор между ними зависит от конкретного проекта и потребностей разработчика.

6. Что нового добавилось в Java 17 в отличии от Java 11?

Java SE 17 (LTS) включает в себя множество новых функций и улучшений по сравнению с Java SE 11. Некоторые из наиболее значимых изменений включают:

1. Новые функции JDK: Java SE 17 включает новые функции JDK, такие как поддержка кодирования с использованием кодовых точек UTF-8, улучшенная поддержка многопоточности и новые инструменты для работы с зависимостями.

2. Поддержка кодирования с использованием кодовых точек UTF-8: Java SE 17 добавляет поддержку кодирования с использованием кодовых точек UTF-8, что облегчает работу с международными языками и символами.

3. Улучшенная поддержка многопоточности: Java SE 17 включает улучшения в области многопоточности, такие как новые методы для работы с потоками и возможность блокировки нескольких потоков одновременно.

4. Новые инструменты для работы с зависимостями: Java SE 17 включает новые инструменты для работы с зависимостями, такие как утилита jlink для создания локальных копий JDK и утилита jde для дебаггинга приложений.

5. Улучшенная безопасность: Java SE 17 включает улучшения в области безопасности, такие как новые механизмы защиты от инъекций кода и улучшенная защита от атак на приложения.

6. Улучшенная производительность: Java SE 17 включает улучшения в области производительности, такие как оптимизации для работы с многопоточностью и новые алгоритмы для работы с коллекциями.

7. Новые возможности для разработки приложений: Java SE 17 включает новые возможности для разработки приложений, такие как поддержка кодирования с использованием кодовых точек UTF-8 и новые инструменты для работы с зависимостями.

В целом, Java SE 17 предлагает множество новых функций и улучшений, которые делают разработку приложений на Java более эффективной и удобной.

Ссылка на github: https://github.com/Kourgeek/java\_2/tree/main

Вывод: в ходе лабораторной работы было реализовано простейшее приложение с помощью фреймворка SpringBoot