Лабораторная работа №1

По предмету Java

Выполнил: Нечман Дмитрий Андреевич,  
 РИМ-220971

Екатеринбург

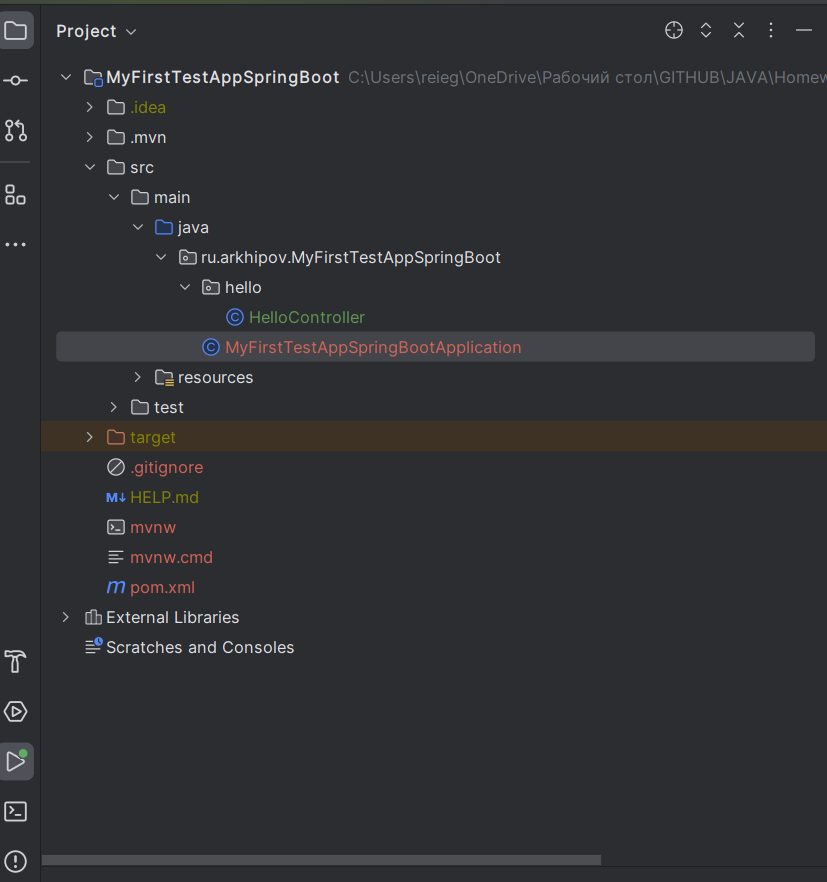
2023

**Цель работы:** Создание простейшего веб-приложения с помощью фреймворка Spring Boot.

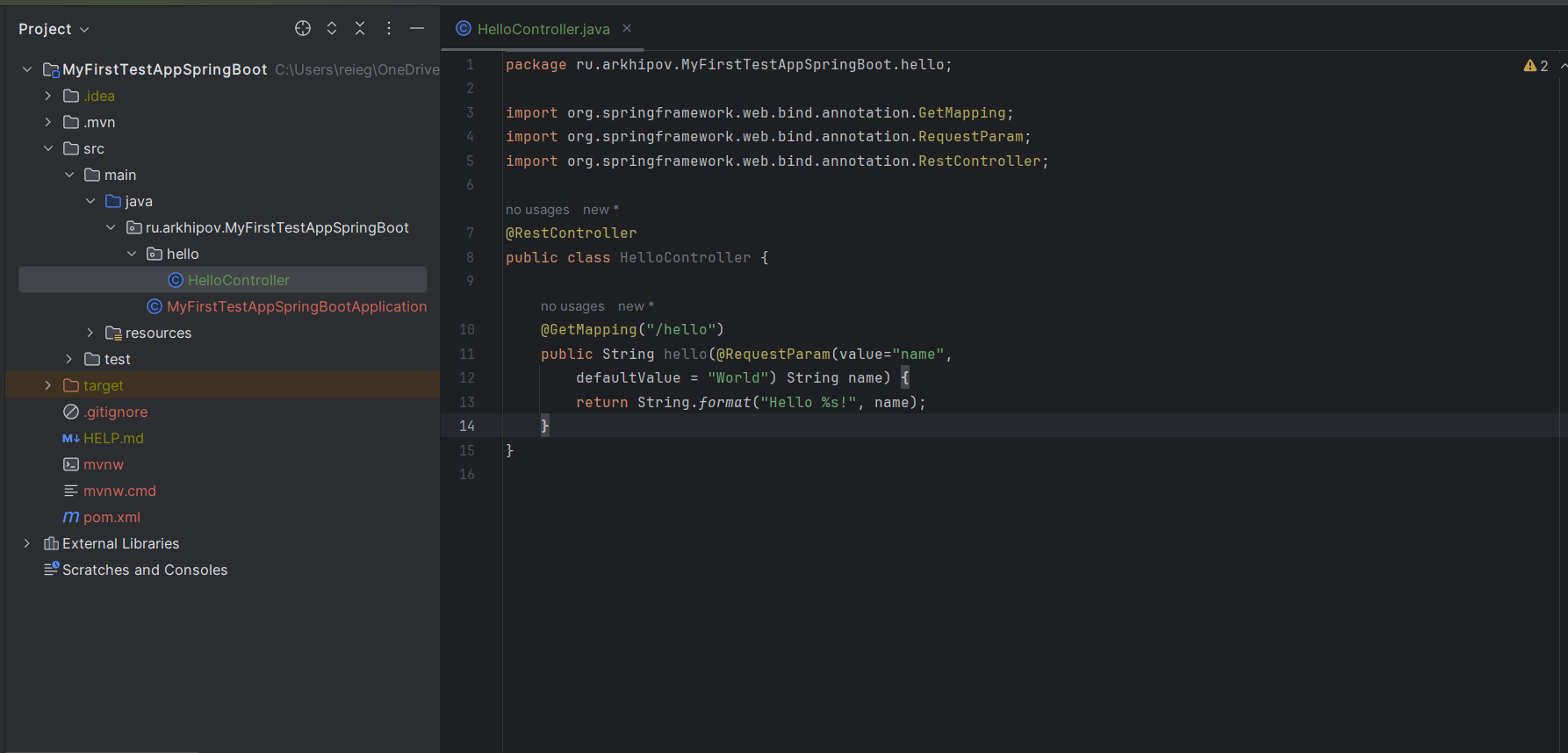
**Задания:**

**I.** **Реализация простейшего Spring Boot Application**

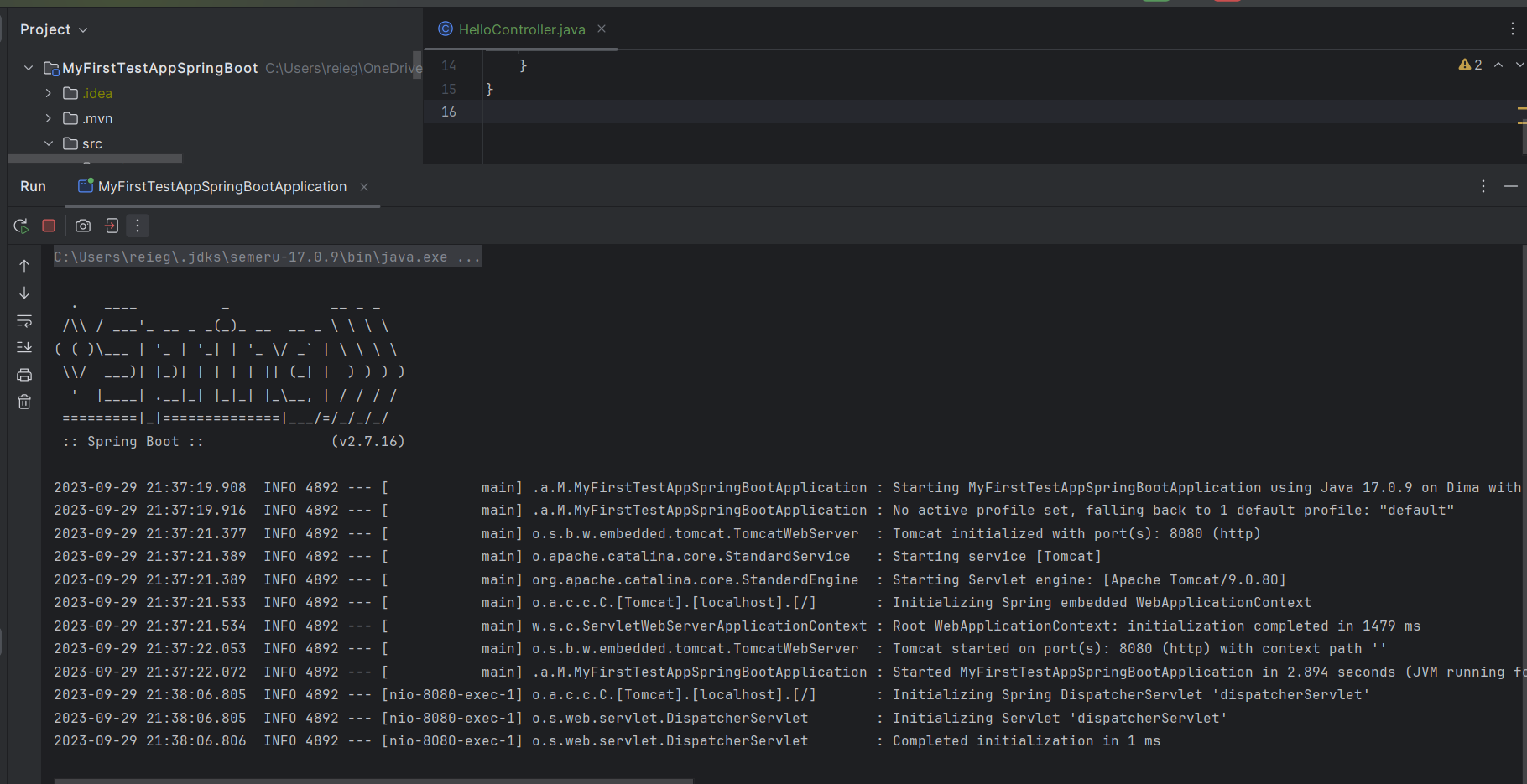
1. Создание конфигурации на сайте https://start.spring.io/



1. Создание класса HelloController



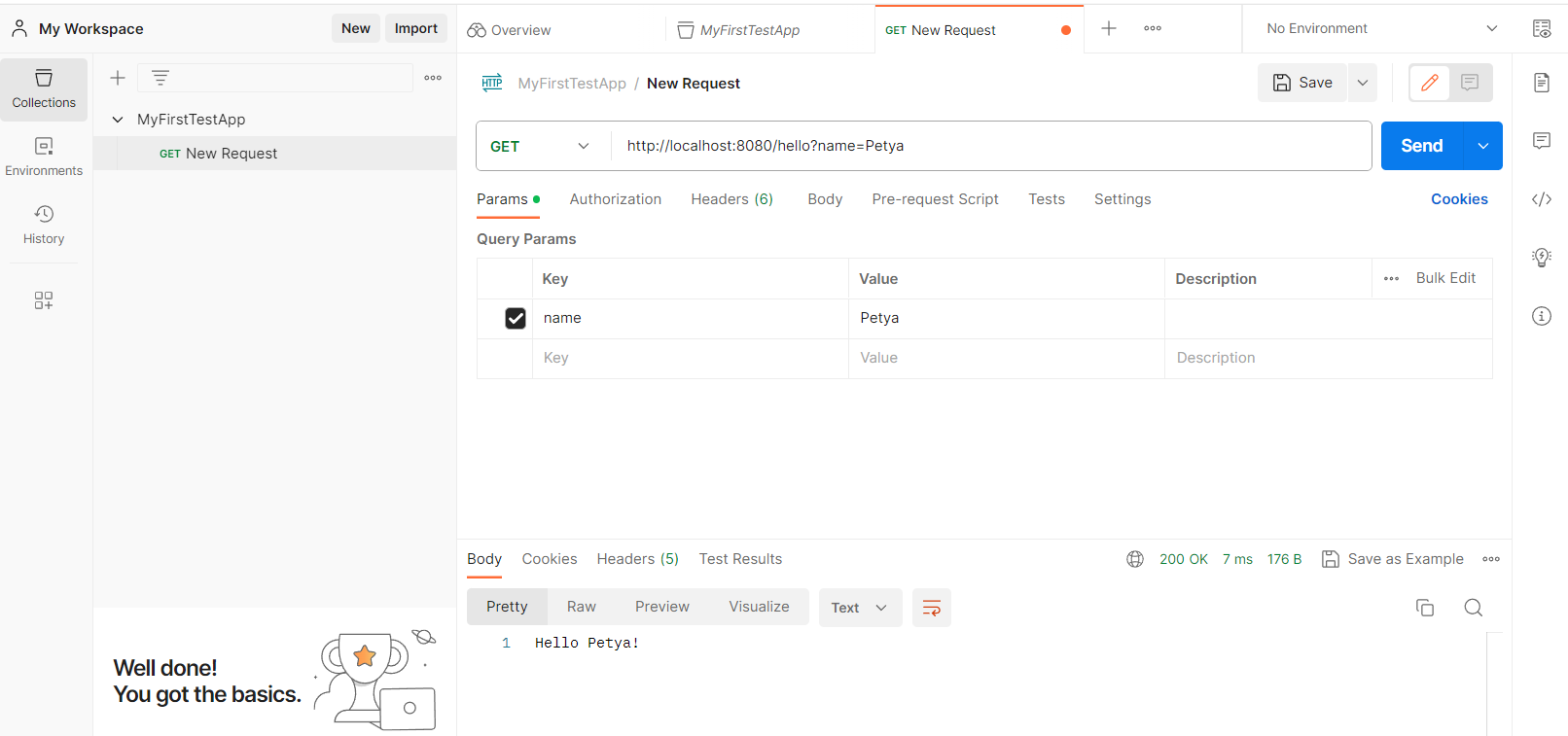
1. Запуск приложения



1. Тестирование в браузере

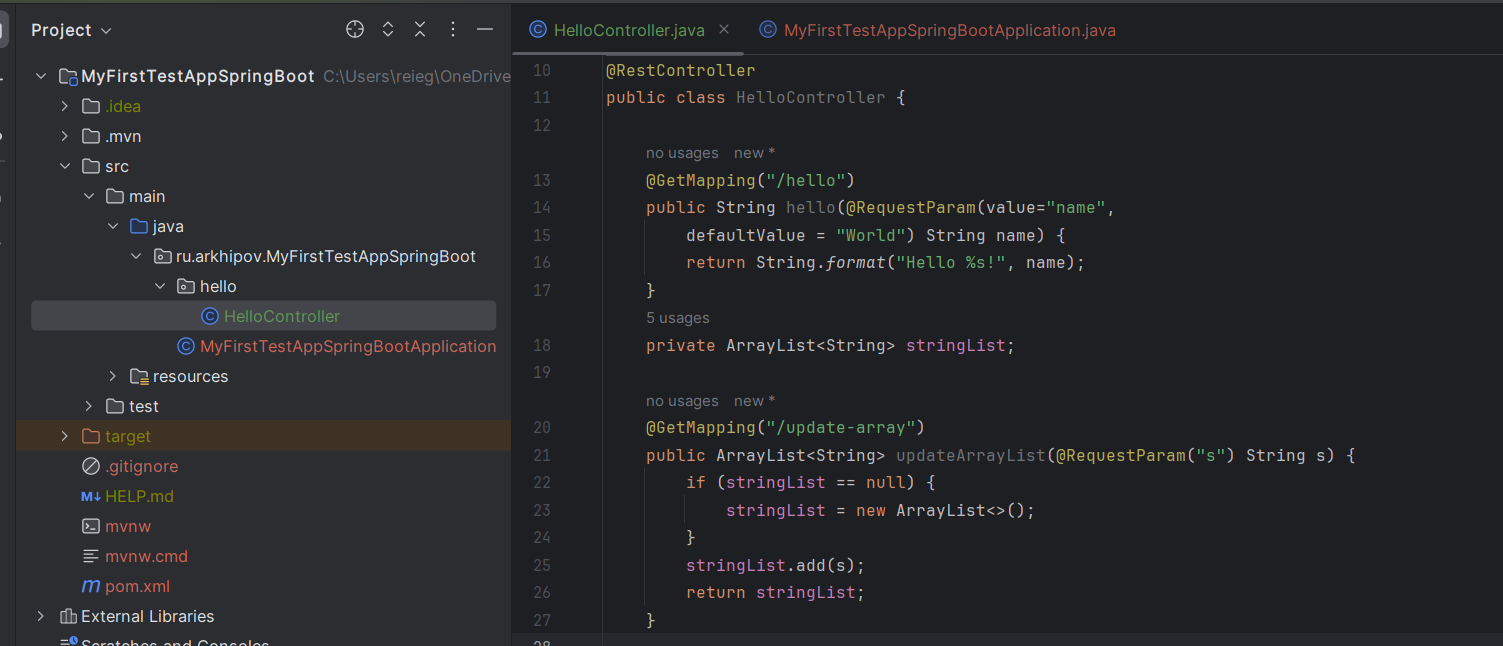


1. Тестирование с помощью Postman

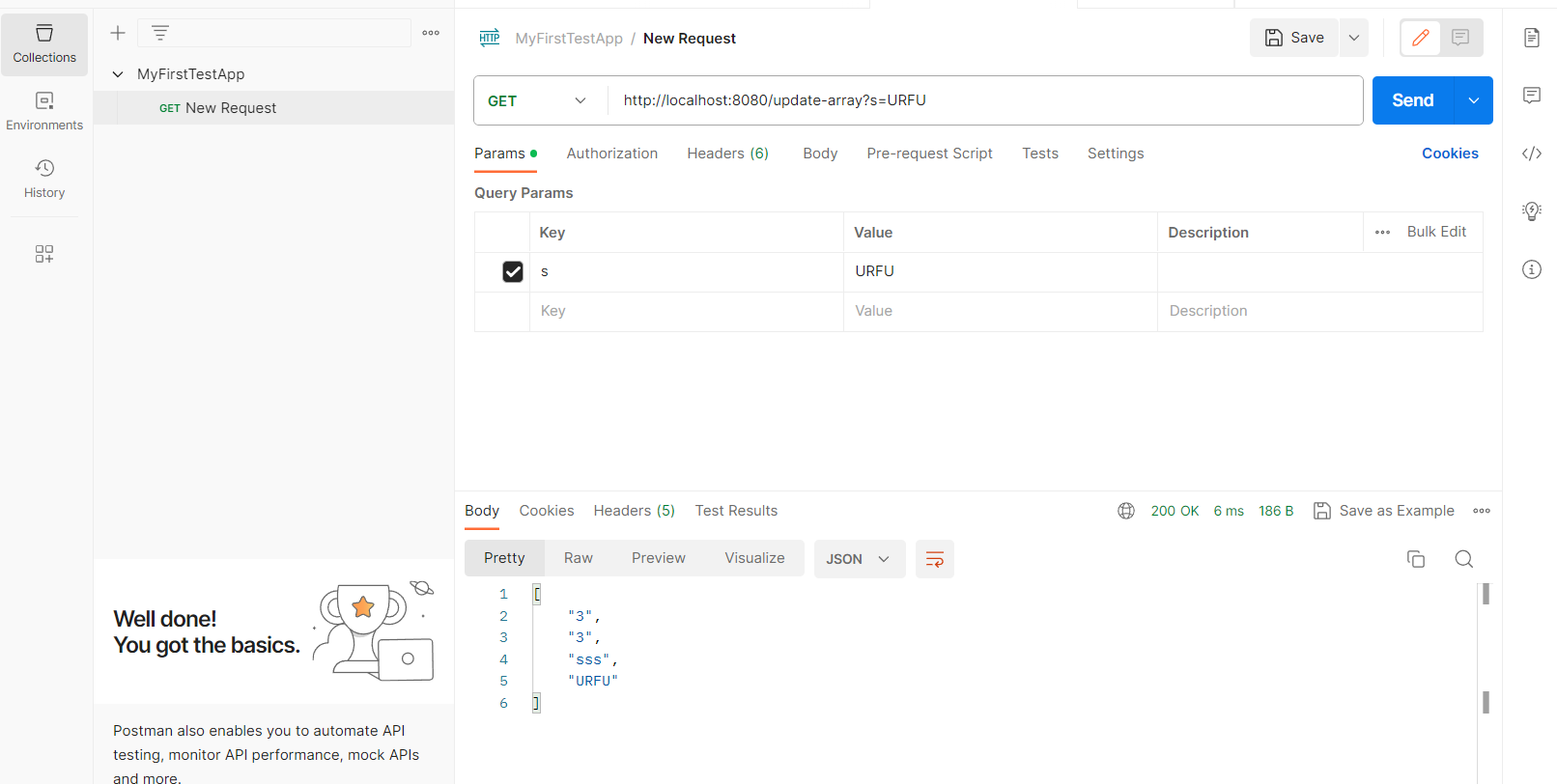


**II.** **Реализация дополнительного функционала**

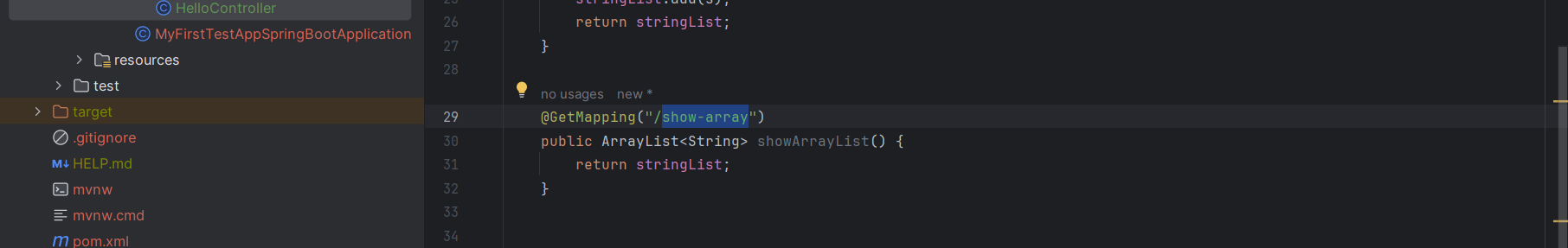
1. Создайте GET-метод updateArrayList(String s), который по url “/update-array” принимает аргумент и записывает его в ArrayList<String >, реализуйте логику - если это первый вызов метода, то создается пустой ArrayList, если он не пустой, то туда записывается значение.

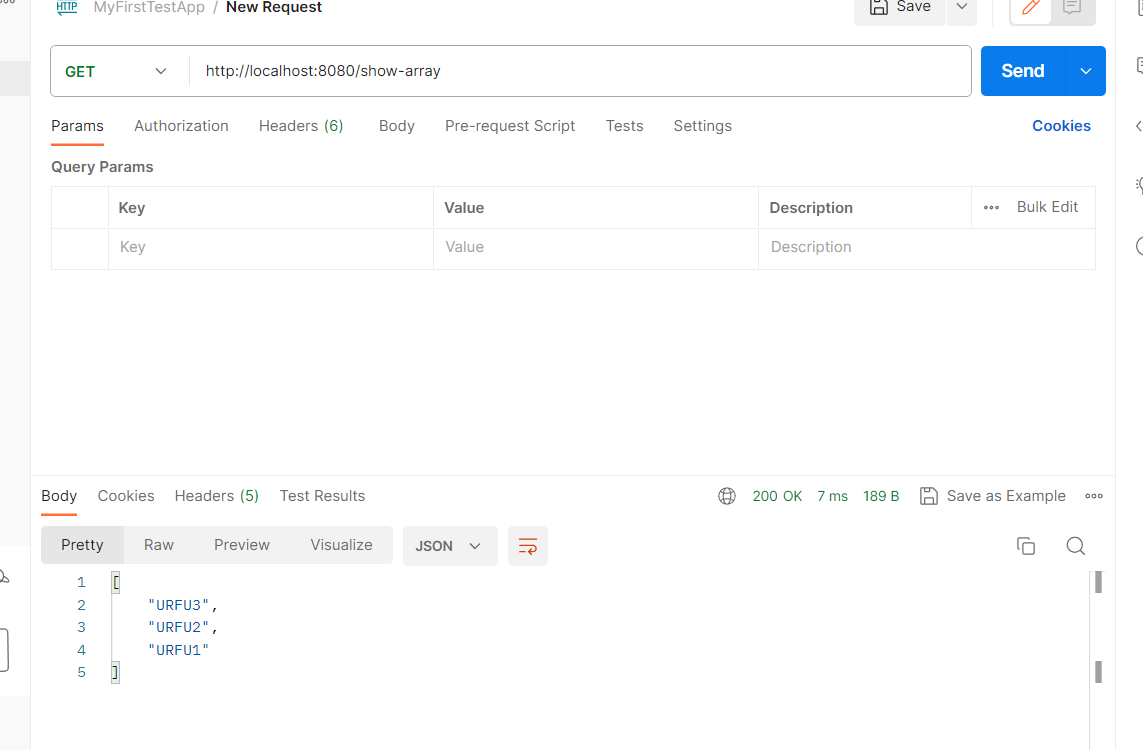


Тестирование через PostMan

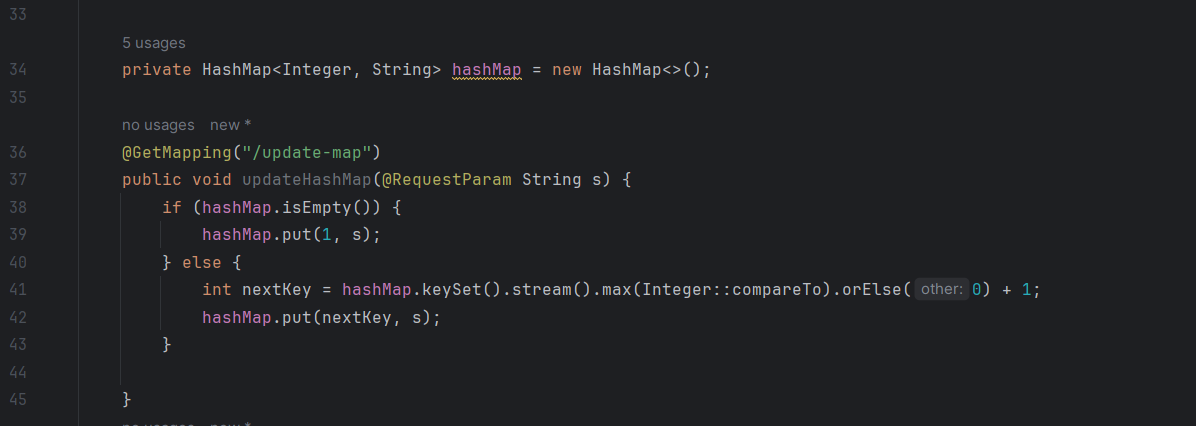


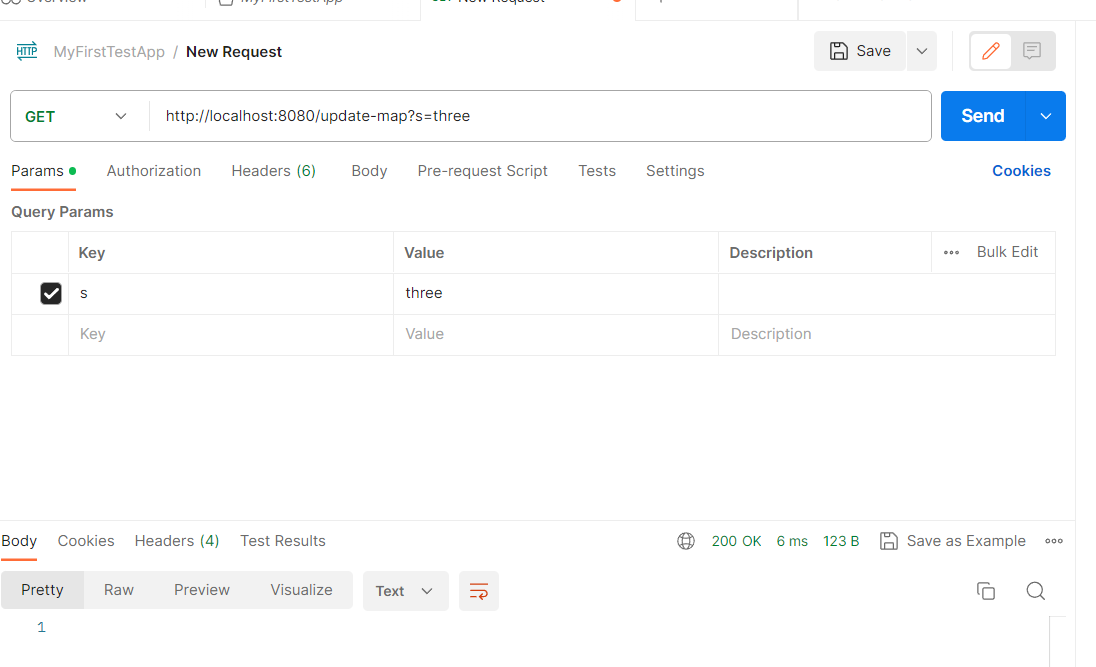
2. Создайте GET метод showArrayList(), который по url “/show-array” возвращается все элементы хранящиеся в ArrayList <String >, созданном в п. 1





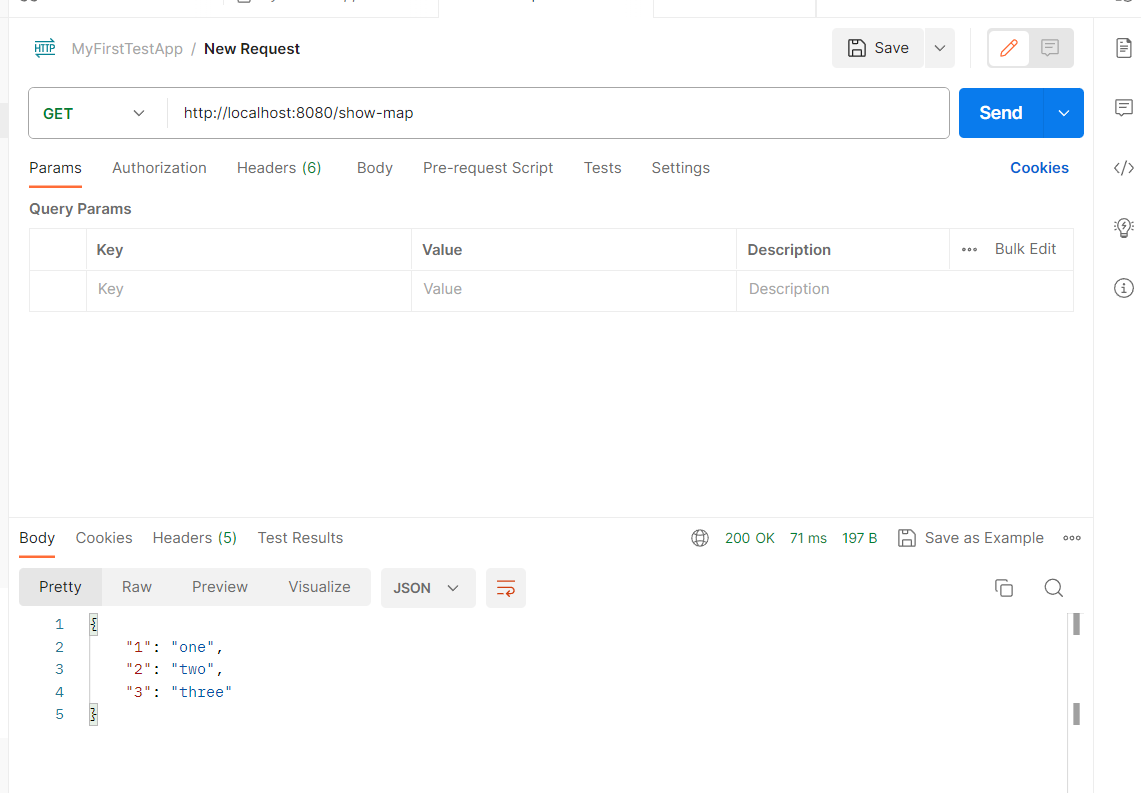
3. Создайте GET-метод updateHashMap(String s), который по url “/update-map” принимает аргумент и записывает его в HashMap< Integer, String>, реализуйте логику - если это первый вызов метода, то создается пустой HashMap, если он не пустой, то туда записывается значение.





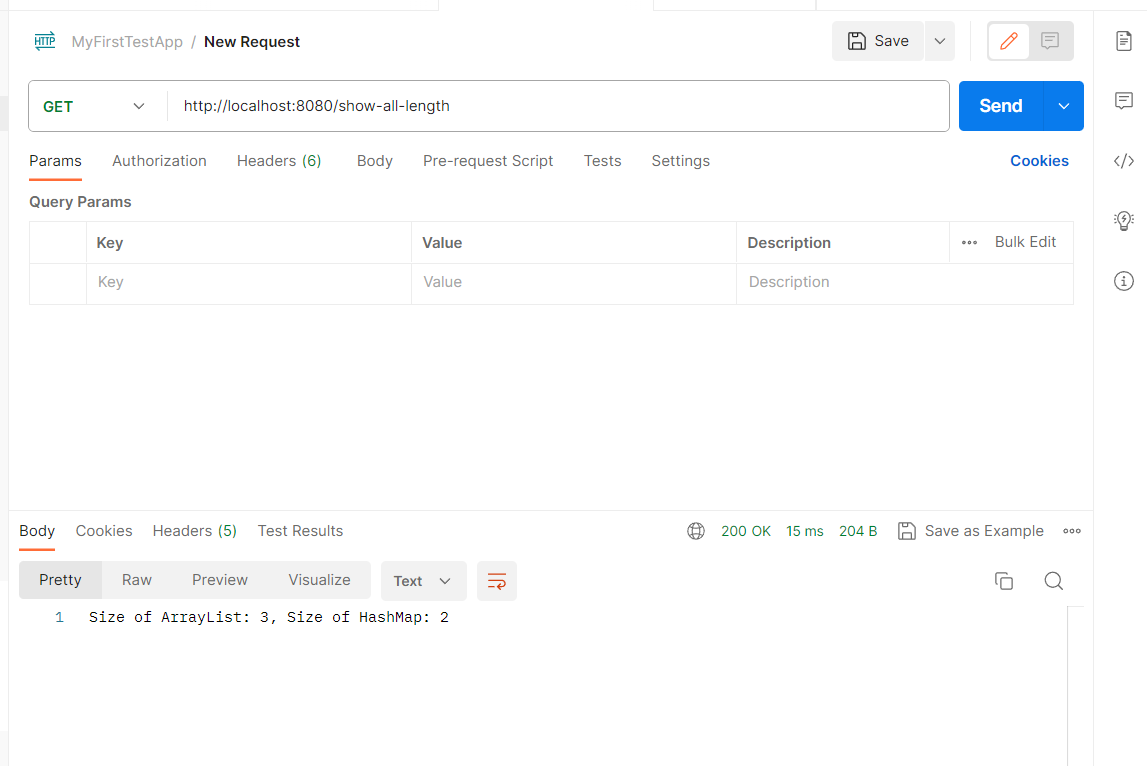
4. Создайте GET метод showHashMap (), который по url “/show-map” возвращается все элементы хранящиеся в HashMap<Integer, String>, созданном в п. 3





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





**Ответы на вопросы:**

2. Чем отличаются языки программирования Java и Kotlin?

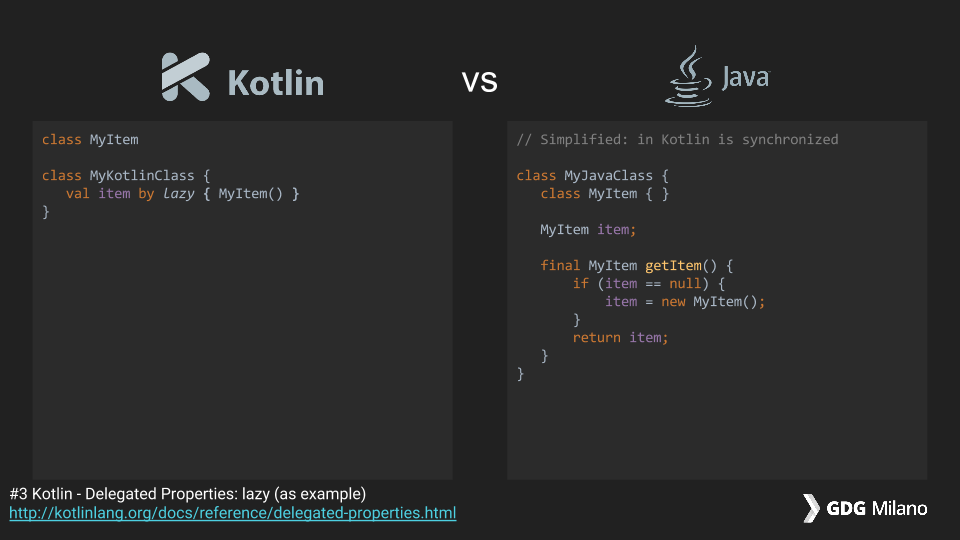
1) Синтаксис: Kotlin имеет более современный и выразительный синтаксис по сравнению с Java. Он использует менее громоздкий и более лаконичный код, что делает его более удобным для разработчиков.

2) Null-безопасность: Kotlin имеет встроенную поддержку null-безопасности, что помогает избежать ошибок, связанных с нулевыми значениями.

3) Расширения функционала: Kotlin предоставляет возможность добавлять новые методы и функции в существующие классы, не изменяя их исходный код. Это называется "расширениями функционала" и позволяет писать более чистый и модульный код. В Java такая возможность отсутствует.

4) Асинхронное программирование: Kotlin имеет синтаксическую поддержку асинхронного программирования с использованием ключевых слов "suspend" и "Coroutine". В Java асинхронное программирование обычно реализуется с использованием коллбэков или библиотек, таких как CompletableFuture или RxJava.

5 Интеграция с Java: Kotlin полностью совместим с Java, поэтому вы можете использовать библиотеки и код на Java в проектах на Kotlin. Это делает переход с Java на Kotlin постепенным и удобным.



4. Чем отличаются языки программирования Kotlin и Groovy

Различия:

1. Kotlin разрабатывался как язык программирования для разработчиков Android, тогда как Groovy был разработан как язык программирования общего назначения.

2. Kotlin может заменить Java на платформе Android, в то время как Groovy используется чаще как дополнительный язык программирования в проектах на Java.

3. Kotlin имеет строгую типизацию, в то время как Groovy имеет динамическую типизацию.

4. Kotlin обеспечивает большую безопасность типов и предотвращает возникновение ошибок времени выполнения за счет статической проверки типов. Groovy, с другой стороны, позволяет более свободное использование типов и не требует такой жесткой типизации.

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

1. Динамические классы: Java 17 представляет новый тип классов, называемых "динамическими классами", которые существуют только во время выполнения программы и могут быть созданы и загружены динамически с использованием API.

2. Улучшения в обработке строк: Java 17 добавляет новые методы и операторы для работы со строками, такие как `stripIndent()` для удаления отступов из многострочных строк и оператор `Text Block` для более читаемого представления многострочных строк.

3. Переполнение кучи (Heap): Java 17 включает улучшения в производительности путем использования эффективных алгоритмов для сжатия объектов и более эффективного использования кучи памяти.

4. Улучшения в обработке потоков: Java 17 предлагает новые методы и улучшения в API потоков для более удобной и гибкой обработки последовательностей данных.

5. Улучшения в манифесте JAR: В Java 17 были добавлены новые атрибуты манифеста JAR, которые позволяют более простую работу с модулями и зависимостями, а также обеспечивают более надежное разрешение конфликтов версий.

6. Улучшения в JEP 394: JEP 394 - это улучшение системы запуска Java, которое введено в Java 11 и внесло изменения в контейнеризацию и процесс запуска приложений Java. В Java 17 были внесены дополнительные улучшения и расширения этого функционала.

**Ссылка на репозиторий:**

https://github.com/NechD/Java\_homework