

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«Национальный исследовательский ядерный университет «МИФИ»  
(НИЯУ МИФИ)**

**Отчет**  
**по результатам выполнения задания**  
**демонстрационного экзамена**

Студент	Колесников Михаил Леонидович
Организация	Национальный исследовательский ядерный университет «МИФИ»
Группа	Б22-534
Дата	31.07.2025

Москва 2025

## Приложение на основе Spring Boot

### Описание функционала приложения

В рамках задания было разработано веб-приложение на основе Spring Boot, предоставляющее REST API для работы с пользователями. Приложение реализует следующий функционал:

1. Получение списка всех пользователей. Эндпоинт: GET /user-api/v1/users. Возвращает JSON-ответ со списком всех пользователей, хранящихся в базе данных.



2. Добавление нового пользователя. Эндпоинт: POST /user-api/v1/users. Принимает JSON-объект с данными нового пользователя и сохраняет его в базе данных.

The screenshot shows a REST client interface. The top bar indicates a POST request to `http://localhost:8080/user-api/v1/users`. Below the bar, tabs for Params, Authorization, Headers (8), Body, Scripts, and Settings are visible. The 'Body' tab is selected, and the content type is set to 'JSON'. The request body is a JSON object:

```
1 {
2   "firstName": "NewUser",
3   "age": 28,
4   "country": "USA"
5 }
```

Below the request body, there are tabs for Body, Cookies, Headers (5), Test Results, and a refresh icon. The 'Body' tab is selected, and the response is shown in JSON format:

```
1 {
2   "id": 6,
3   "firstName": "NewUser",
4   "age": 28,
5   "country": "USA"
6 }
```

3. Получение пользователей по возрасту. Эндпоинт: GET /user-api/v1/additional-info?age=X. Возвращает JSON-ответ со списком пользователей, возраст которых больше или равен указанному значению X, отсортированных по имени в алфавитном порядке

The screenshot shows a REST client interface. The top bar indicates a GET request to `http://localhost:8080/user-api/v1/additional-info?age=25`. Below the bar, tabs for Params, Authorization, Headers (6), Body, Scripts, and Settings are visible. The 'Body' tab is selected, and the content type is set to 'JSON'. The response body is a JSON array of three user objects:

```
1 [
2   {
3     "id": 2,
4     "firstName": "Ivan",
5     "age": 30,
6     "country": "RUSSIA"
7   },
8   {
9     "id": 1,
10    "firstName": "John",
11    "age": 25,
12    "country": "USA"
13  },
14  {
15    "id": 6,
16    "firstName": "NewUser",
17    "age": 28,
18    "country": "USA"
19  }
20 ]
```

## Технические детали реализации

Стек технологий: Java, Spring Boot, Spring Web, Spring Data JPA, H2 Database.

Архитектура: Приложение разделено на слои:

1. Контроллер (UserController) — обработка HTTP-запросов.
2. Сервис (UserService) — бизнес-логика.
3. Репозиторий (UserRepository) — взаимодействие с базой данных.
4. Используется in-memory H2 database с таблицей users. При запуске приложения автоматически создаются тестовые данные (5 пользователей, по одному для каждой страны).

Сущности:

- Класс User с полями: id, firstName, age, country.
- Enum Country со значениями: USA, RUSSIA, GERMANY, FRANCE, JAPAN.

Ссылка на репозиторий проекта:

<https://github.com/MichaelKolesnikov/DemonstrationExamDD>

## Заключение

В ходе выполнения задания было успешно разработано Spring Boot приложение, реализующее REST API для управления пользователями. Все требования задания выполнены:

1. Реализованы все указанные эндпоинты.
2. Использованы Spring Web и Spring Data JPA.
3. Данные хранятся в H2 database с автоматической инициализацией тестовых данных.
4. Архитектура приложения соответствует стандартным практикам (контроллер-сервис-репозиторий).

Приложение готово к дальнейшему расширению функционала, например, добавлению аутентификации, валидации данных или подключению внешней базы данных.