

ОТЧЕТ
ПО РЕЗУЛЬТАТАМ ВЫПОЛНЕНИЯ ЗАДАНИЯ
ДЕМОНСТРАЦИОННОГО ЭКЗАМЕНА
Вариант №1

Студент: Почернин Владислав Сергеевич

Организация: НИЯУ МИФИ

Группа: М24-535

Дата: 12.07.2025

Содержание

1	Задание	2
1.1	Постановка задачи	2
1.2	Обязательные требования	2
1.3	Все остальные детали реализации - на усмотрение разработчика	2
2	Приложение на основе Spring Boot	4
2.1	Инициализация проекта	4
3	Заключение	6

1 Задание

1.1 Постановка задачи

Создать небольшое Java web-приложение на основе Spring Boot со следующим функционалом:

- 1) При обращении по эндпоинту `user-api/v1/users` (GET метод) приложение возвращает json-ответ со списком всех доступных пользователей.
- 2) При обращении по эндпоинту `user-api/v1/users` (POST метод) происходит добавление нового пользователя с заданными параметрами.
- 3) При обращении по эндпоинту `user-api/v1/additional-info` (GET метод) приложение возвращает json-ответ со списком всех пользователей, возраст которых больше либо равен заданному, отсортированный по имени (`firstName`) в алфавитном порядке. Заданный возраст передается в строке запроса в виде параметра (например, `user-api/v1/additional-info?age=18`), при этом данный эндпоинт `user-api/v1/additional-info` должен быть универсальным, т.е. работать для любого значения возраста.

1.2 Обязательные требования

- Пользователь описывается сущностью `User` с полями `Long id`, `String firstName`, `Integer age`, `Country country`, `Country` - это enum с названиями стран с некоторым набором возможных значений (заполнить минимум 5 произвольными странами).
- Данные всех пользователей хранятся в базе данных в таблице `users`, добавление новых пользователей также происходит в эту таблицу.
- Java, Spring Boot.
- Модули Spring WEB, Spring Data JPA.
- Приложение разбито на слои: репозиторий, сервис, (REST-)контроллер (использовать соответствующие Spring аннотации).
- Для взаимодействия с БД подключить к проекту и использовать in-memory H2 database.
- Функционал, описанный в пунктах 1, 2 и 3, реализовать через JPA репозиторий для сущности `User` (без написания SQL запросов).
- При запуске приложения должна создаваться таблица `users` минимум с 5 записями - по одной для каждой страны (допускается реализовать при помощи SQL скриптов).

1.3 Все остальные детали реализации - на усмотрение разработчика

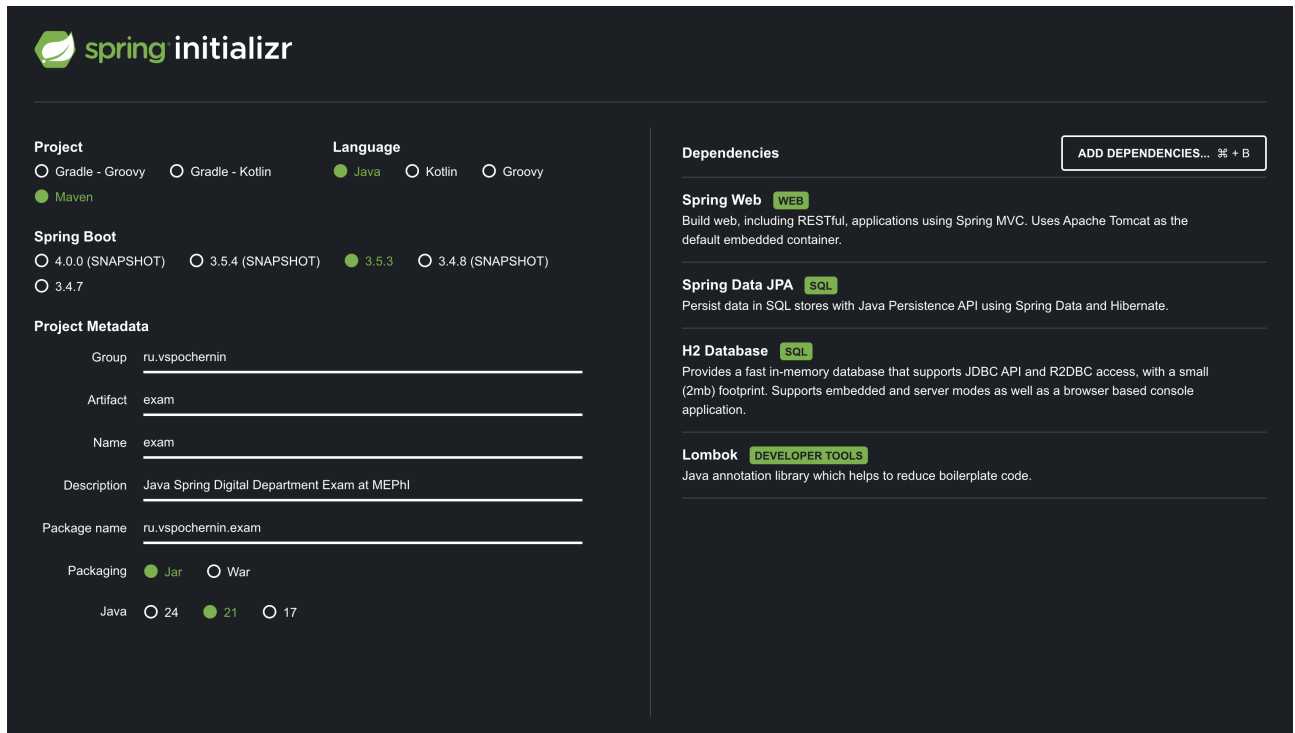
- Пояснение: под эндпоинтом понимается следующее - по умолчанию локально запущенное Spring Boot приложение работает на `http://localhost:8080/`, соответственно, для добавления нового пользователя необходимо отправить POST запрос (информация о новом пользователе указывается в теле запроса) на `http://localhost:8080/user-api/v1/users`.

- Для отправки запросов к приложению (для локального тестирования работы приложения) можно использовать любой REST клиент (Insomnia, Postman, IntelliJ IDEA Http Client, ...).
- Пояснение при использовании Community версии IntelliJ IDEA: для создания проекта использовать ссылку <https://start.spring.io/>. Там же произвести выбор необходимых зависимостей. Затем скачать сгенерированный проект и открыть его при помощи IntelliJ IDEA Community.

2 Приложение на основе Spring Boot

2.1 Инициализация проекта

Для инициализации проекта воспользуемся сервисом <https://start.spring.io/>.



The screenshot shows the Spring Initializr web application interface. It is divided into two main sections: Project Metadata and Dependencies.

Project Metadata:

- Project:** ☐ Gradle - Groovy, ☐ Gradle - Kotlin, ☒ Maven
- Language:** ☒ Java, ☐ Kotlin, ☐ Groovy
- Spring Boot:** ☐ 4.0.0 (SNAPSHOT), ☐ 3.5.4 (SNAPSHOT), ☒ 3.5.3, ☐ 3.4.8 (SNAPSHOT), ☐ 3.4.7
- Project Metadata:**
 - Group: ru.vspochernin
 - Artifact: exam
 - Name: exam
 - Description: Java Spring Digital Department Exam at MEPhI
 - Package name: ru.vspochernin.exam
 - Packaging: ☒ Jar, ☐ War
 - Java: ☐ 24, ☒ 21, ☐ 17

Dependencies:

- Spring Web (WEB):** Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
- Spring Data JPA (SQL):** Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.
- H2 Database (SQL):** Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.
- Lombok (DEVELOPER TOOLS):** Java annotation library which helps to reduce boilerplate code.

Buttons: ADD DEPENDENCIES... % + B

Рис. 1: Инициализация проекта

Мы будем использовать Java версии 21, сборщик Maven, а также актуальную на момент написания отчета версию Spring Boot - 3.5.3.

Из зависимостей были выбраны:

- Spring Web - для создания сервера, который сможет отвечать на HTTP запросы.
- Spring Data JPA - для взаимодействия с нашей H2 базой данных с помощью объектно-реляционного отображения.
- H2 Database - драйвер для работы H2 базы данных.
- Lombok - вспомогательный инструмент для уменьшения количества boilerplate кода.

3 Заключение