|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №7**

по дисциплине «Разработка серверных частей интернет-ресурсов»

**Тема практической работы:**

**Студент группы** ИКБО-33-21 Май Ван Туан

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель практической работы** преподаватель Волков М.Ю.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Москва 2023

СОДЕРЖАНИЕ

[**1. Цель работы** 3](#_Toc145190172)

[**2. Ход работы** 3](#_Toc145190173)

[**3. Ответы на вопросы** 5](#_Toc145190174)

## Постановка задачи

Студенту предлагается дополнить задание шестой практики по Java. Нужно создать приложение на микросервисной архитектуре, где один сервис — это приложение шестой практики, а второй сервис — это сервис авторизации.

Сервис авторизации должен быть написан с помощью Spring Security и содержать JWT token. В качестве СУБД во втором сервисе должен быть использован Redis. Должны быть созданы три роли: USER, SELLER, ADMINISTRATOR.

USER не имеет доступ к какому-либо методу, который связан с редактированием или просмотром информации по другому пользователю, не имеет возможности добавлять или удалять товары.

SELLER имеет те же ограничения, что и USER, за исключением, что продавец может добавлять и удалять свои товары.

ADMINISTRATOR не имеет каких либо ограничений.

В сервис с приложением маркетплейса добавить бизнес логику, позволяющую пользователь с ролью USER получить роль SELLER.

Практика должна запускаться с помощью docker-compose. Каждый микросервис должен запускаться в отдельном потоке.

## Ход работы

Для начала создадим все нужные файлы для создания конфигурации серверного ПО (рис. 1).

A screenshot of a computer

Description automatically generated

1. Структура образа

Содержимое всех необходимых файлов показано на рисунках 2-8.

A computer screen shot of a black screen

Description automatically generated

1. Файл application.properties

A black rectangular object with a black border

Description automatically generated

1. Файл Dockerfile

A screenshot of a computer program

Description automatically generated

1. Файл docker-composer.yml

A screen shot of a computer program

Description automatically generated

1. Файл JwtAuthenticationFilter

A screen shot of a computer program

Description automatically generated

1. Файл SecurityConfiguration

A screen shot of a computer program

Description automatically generated

1. Сборка User

A screenshot of a computer program

Description automatically generated

1. Сборка UseController

A screenshot of a computer program

Description automatically generated

1. Сборка AuthenticaitonController

Далее была протестирована функциональность серверного Postmen (Рисунок 8 -10).

A screenshot of a computer

Description automatically generated

1. Signup

A screenshot of a computer

Description automatically generated

1. Signin

A screenshot of a computer

Description automatically generated

1. Authentication with role

## Ответы на вопросы:

* **Spring Security**

Spring Security - это фреймворк безопасности для приложений на платформе Java. Он предоставляет мощные и гибкие средства аутентификации и авторизации.

Он интегрируется с различными фреймворками Spring, обеспечивая защиту от множества атак, таких как атаки на сеансы, кросс-сайтовый скриптинг и другие.

* **Что такое JWT Token и как он работает**

JWT (JSON Web Token) - это компактный, самосодержащийся формат передачи информации между сторонами в виде объекта JSON.

Он состоит из трех частей: заголовка, полезной нагрузки и подписи. Используется для аутентификации и обмена информацией между клиентом и сервером.

* **Микросервисная архитектура. Плюсы и минусы перед монолитной системой**

Плюсы: легкость масштабирования, независимость служб, более гибкий и быстрый развитие, лучшая устойчивость к отказам.

Минусы: сложность управления, повышенные затраты на инфраструктуру, проблемы с согласованностью данных между службами.

* **Варианты масштабируемости**

Вертикальная масштабируемость (повышение мощности одного сервера).

Горизонтальная масштабируемость (добавление новых серверов).

* **Что такое вертикальная масштабируемость**

Увеличение мощности сервера путем добавления ресурсов (памяти, процессора и т. д.).

* **Что такое горизонтальная масштабируемость**

Увеличение мощности путем добавления новых серверов в сеть.

* **Объяснить принцип сбора микросервисной архитектуры в Docker**

Docker позволяет упаковывать приложения и их зависимости в контейнеры, обеспечивая консистентность в различных окружениях и облегчая развертывание.

* **NoSQL DB. Почему в сервисе авторизации удобнее использовать Redis**

В сервисе авторизации удобно использовать Redis из-за его быстродействия и возможности хранения ключ-значение, что подходит для хранения токенов сессий.

* **Многопотчность — что это такое. Как оно работает в Java**

Это концепция выполнения нескольких потоков одновременно. В Java, многопоточность реализуется через классы, такие как Thread и Runnable.

* **Класс Thread, расскажите про его методы**

start(): запускает новый поток.

run(): содержит код, который будет выполнен в потоке.

sleep(ms): заставляет поток заснуть на указанное количество миллисекунд.