МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Отчет по выполнению практической работы №3

Тема:

СИСТЕМЫ СБОРКИ

Дисциплина: «Технология разработки программных приложений»

Выполнил студент: Враженко Д.О.

Группа: ИКБО-50-23

Вариант: 6

Москва – 2025

ОГЛАВЛЕНИЕ

ЦЕЛЬ РАБОТЫ.....3
ХОД РАБОТЫ.....4
КОНТРОЛЬНЫЕ ВОПРОСЫ.....6
ВЫВОД.....7

ЦЕЛЬ РАБОТЫ

Цель работы:

Знакомство с системой сборки Gradle. Возможности gradle. Управление зависимостями.

Задание для выполнения:

Для выполнения необходимо клонировать (или форкнуть) git-репозиторий согласно варианту, и выполнить следующие задания:

1. Найти отсутствующую зависимость и указать ее в соответствующем блоке в build.gradle, чтобы проект снова начал собираться.

2. В некоторых классах поправить имя пакета.

3. Собрать документацию проекта, найти в ней запросы состояния и сущности по идентификатору.

4. Собрать jar со всеми зависимостями (так называемый UberJar), после чего запустить приложение. По умолчанию, сервер стартует на порту 8080.

5. Запросить состояние запущенного сервера (GET запрос по адресу http://localhost:8080).

6. Запросить сущность по идентификатору (GET запрос по адресу: http://localhost:8080/сущность/идентификатор).

Идентификатором будут 3 последних цифры в серийном номере вашего студенческого билета.

7. В задаче shadowJar добавить к jar-файлу вашу фамилию.

8. Выполнить задачу checkstyleMain. Посмотреть сгенерированный отчет. Устранить ошибки оформления кода.

Вариант для выполнения:

6) репозиторий: https://github.com/rtu-mirea/trpp-second-6, сущность ru.mirea.entity.Message

ХОД РАБОТЫ

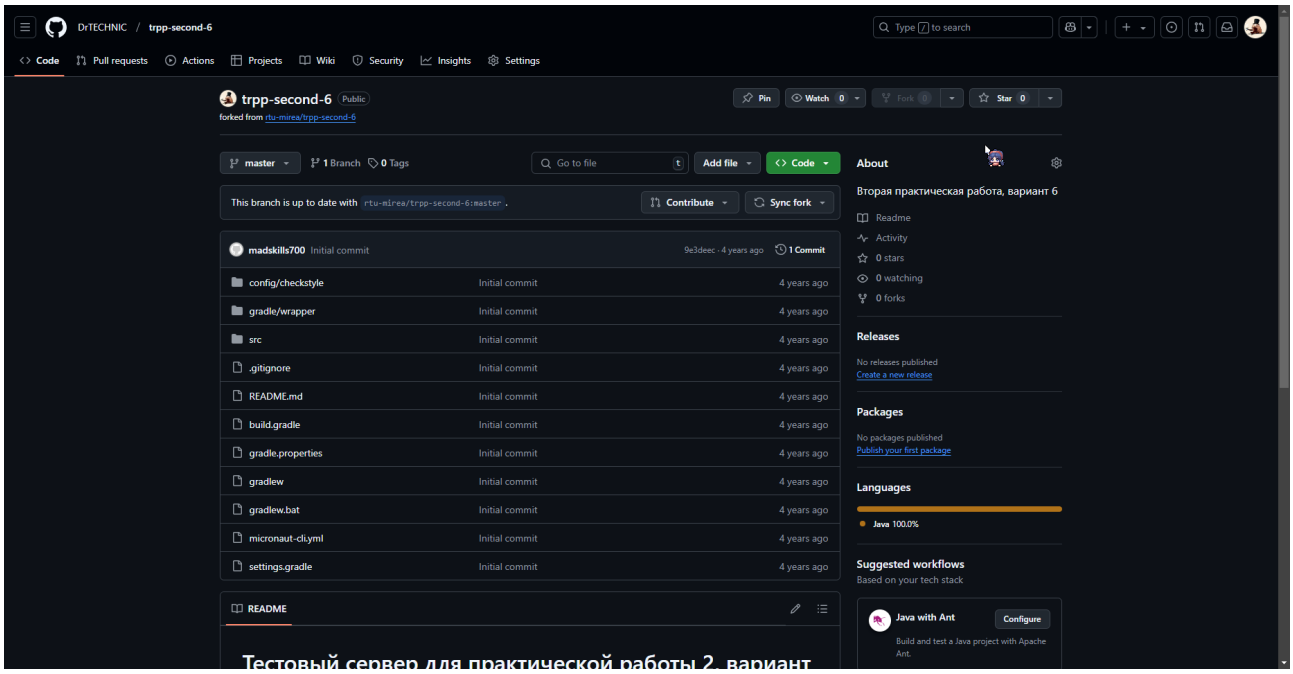

Рисунок 1 - Форк репозитория

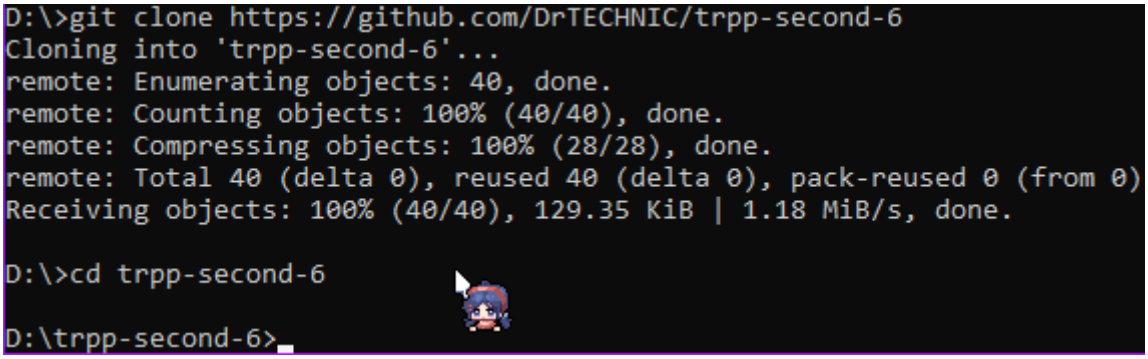

Рисунок 2 - Клонирование репозитория

```
dependencies {
    annotationProcessor 'org.projectlombok:lombok:1.18.18'
    compileOnly 'org.projectlombok:lombok:1.18.18'

    implementation("io.micronaut:micronaut-runtime")
    implementation("io.micronaut:micronaut-validation")
    implementation("io.micronaut:micronaut-http-client")
    implementation("javax.annotation:javax.annotation-api")
    implementation('org.apache.logging.log4j:log4j-core:2.17.1')
    runtimeOnly("org.apache.logging.log4j:log4j-api:2.12.1")
    runtimeOnly("org.apache.logging.log4j:log4j-slf4j-impl:2.12.1")
    implementation("com.opencsv:opencsv:5.7.1") // new
}
```

Рисунок 3 - Добавленная зависимость

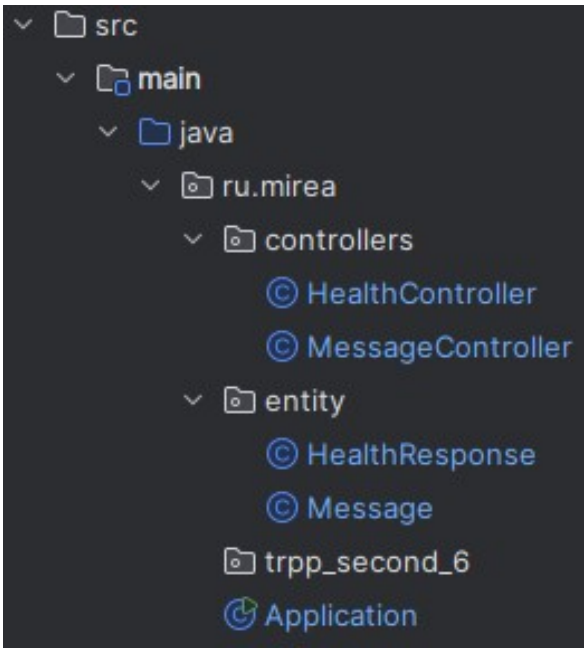

Рисунок 4 - Изменение названия пакета

Method Detail

getMessages

`@Get public io.micronaut.http.HttpResponse<java.util.List<Message>> getMessages()`

Получить список сообщений.

Returns:
список сообщений

findById

`@Get("/{id}") public io.micronaut.http.HttpResponse<Message> findById(java.lang.Long id)`

Найти сообщение по идентификатору.

Parameters:
id - идентификатор сообщения

Returns:
Сообщение, если есть, иначе 404 ошибка

Рисунок 5 - Запрос состояния и сущности по идентификатору

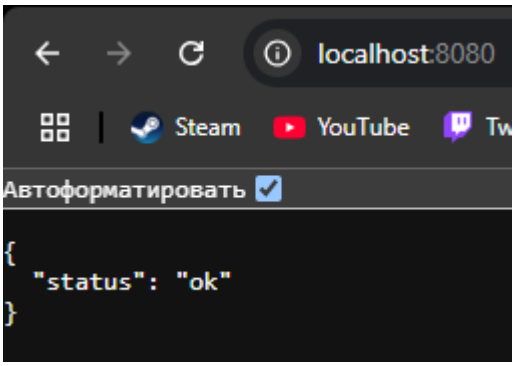

Рисунок 6 - Статус сервера

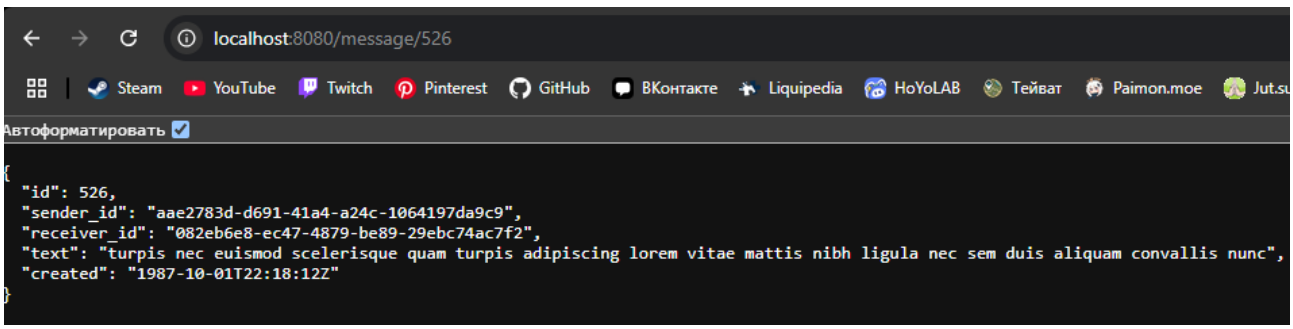

Рисунок 7 - Запрос сущности по идентификатору

```
shadowJar {
    archivesBaseName = "${project.name} - Vrazhenko"
    libsDirName = "${project.name}"
    classifier('')
}
```

Рисунок 8 - Фамилия в shadowJar

```
15:55:52: Executing 'checkstyleMain'...

> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
> Task :checkstyleMain

BUILD SUCCESSFUL in 2s
3 actionable tasks: 1 executed, 2 up-to-date
15:55:55: Execution finished 'checkstyleMain'.
```

Рисунок 9 - CheckstyleMain

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Чем компиляция отличается от сборки?

Компиляция — преобразование исходного кода в машинный код или байт-код. Сборка — процесс создания готового приложения, включающий компиляцию, линковку, упаковку и другие этапы.

2. Что такое система сборки?

Система сборки — инструмент для автоматизации процессов компиляции, тестирования, упаковки и развертывания программного обеспечения.

5. Что такое gradle?

Gradle — система сборки с поддержкой Groovy и Kotlin DSL, используемая для автоматизации сборки проектов.

6. Что такое maven?

Maven — система сборки и управления зависимостями, использующая XML для конфигурации.

15. Что такое UberJar? При помощи какой задачи его собрать?

UberJar — JAR-файл, содержащий все зависимости и классы приложения. Собирается с помощью задачи shadowJar в Gradle или maven-shade-plugin в Maven.

19. Что такое аннотация в Java?

Аннотация в Java — метаданные, добавляемые в код для предоставления информации компилятору, runtime или другим инструментам.

ВЫВОД

В ходе выполнения практической работы была изучена система сборки Gradle, её основные возможности и принципы работы. Рассмотрены ключевые аспекты управления зависимостями, включая подключение внешних библиотек и использование репозиториев. Практическое применение Gradle позволило освоить базовые команды, структуру проекта и механизмы автоматизации сборки. Полученные знания могут быть использованы для эффективного управления зависимостями и упрощения процесса разработки в будущих проектах.