



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных
технологий (МОСИТ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 3

по дисциплине

«Технология разработки программных приложений»

Тема: «Системы сборки»

Выполнил студент группы ИКБО-13-22

Тринеев П.С

Принял доцент

Петренко А.А.

Практическая работа выполнена

✓

«06» 04 2024 г.

(подпись студента)

«Зачтено»

✓

«06» 04 2024 г.

(подпись руководителя)

Москва 2024

Содержание

1 Задание для выполнения	2
1.1 Найти отсутствующую зависимость и указать ее в соответствующем блоке в build.gradle, чтобы проект снова начал собираться.....	2
1.2 В некоторых классах поправить имя пакета	2
.....	2
1.3 Собрать документацию проекта, найти в ней запросы состояния и сущности по идентификатору	3
.....	3
1.4 Собрать jar со всеми зависимостями (так называемый UberJar), после чего запустить приложение. По умолчанию, сервер стартует на порту 8080.	4
1.5 Запросить состояние запущенного сервера (GET запрос по адресу http://localhost:8080) и запросить сущность по идентификатору (GET запрос по адресу: http://localhost:8080/сущность/идентификатор)	4
1.6 В задаче shadowJar добавить к jar-файлу вашу фамилию	4
1.7 Выполнить задачу checkstyleMain. Посмотреть сгенерированный отчет. Устранить ошибки оформления кода.	5
2 Ответы на вопросы.....	6
Вывод.....	7

1 Задание для выполнения

Вариант №14. Номер места в списке группы 29.

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

```
dependencies {
    annotationProcessor 'org.projectlombok:lombok:1.18.18'
    compileOnly 'org.projectlombok:lombok:1.18.18'
    implementation 'com.opencsv:opencsv:5.5'
    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:2.12.1")
    runtimeOnly("org.apache.logging.log4j:log4j-api:2.12.1")
    runtimeOnly("org.apache.logging.log4j:log4j-slf4j-impl:2.12.1")
}
```

Рисунок 1 – Отсутствующая зависимость

1.2 В некоторых классах поправить имя пакета

```
package ru.mirea.trpp_second_6.controllers;
import ru.mirea.trpp_second_6.entity.Message;
import com.opencsv.bean.CsvToBeanBuilder;
import io.micronaut.http.HttpResponse;
import io.micronaut.http.annotation.Controller;
import io.micronaut.http.annotation.Get;

import java.io.InputStreamReader;
import java.util.List;
import java.util.Optional;
```

Рисунок 2 – Изменения в файле MessageController.java

```
application {
    mainClass.set("ru.mirea.trpp_second_6.Application")
}
```

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

```
PS C:\Users\lenovo\IdeaProjects\trpp-second-6> ./gradlew javadoc  
  
BUILD SUCCESSFUL in 4s  
3 actionable tasks: 1 executed, 2 up-to-date
```

Рисунок 3 – Сбор документации проекта

Method Detail	
getMessages	<pre>@Get public io.micronaut.http.HttpResponse<java.util.List<Message>> getMessages()</pre> <p>Получить список сообщений.</p> <p>Returns: список сообщений</p>
findById	<pre>@Get("/{id}") public io.micronaut.http.HttpResponse<Message> findById(java.lang.Long id)</pre> <p>Найти сообщение по идентификатору.</p> <p>Parameters: id - идентификатор сообщения</p> <p>Returns: Сообщение, если есть, иначе 404 ошибка</p>

Рисунок 4 – Запросы состояния и сущности

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

```
PS C:\Users\lenovo\IdeaProjects\trpp-second-6> ./gradlew shadowJar

BUILD SUCCESSFUL in 2s
3 actionable tasks: 3 up-to-date
PS C:\Users\lenovo\IdeaProjects\trpp-second-6> java -jar .\build\trpp-second-6\trpp-second-6-0.1.jar
WARNING: sun.reflect.Reflection.getCallerClass is not supported. This will impact performance.
2024-03-10 22:27:05,161 main DEBUG null null initializing configuration XmlConfiguration[location=jar:file:/C:/Users/lenovo/IdeaProjects/trpp-second-6/build/trpp-second-6/trpp-second-6-0.1.jar!/log4j2.xml]
2024-03-10 22:27:05,169 main DEBUG Installed 1 script engine
```

Рисунок 5 – Сборка Uberjar и запуск приложения

1.5 Запросить состояние запущенного сервера (GET запрос по адресу <http://localhost:8080>) и запросить сущность по идентификатору (GET запрос по адресу: <http://localhost:8080/сущность/идентификатор>)

```
C:\Users\lenovo>curl http://localhost:8080
{"status":"ok"}
C:\Users\lenovo>curl http://localhost:8080/message/393
{"id":393,"sender_id":"6d1ca488-d5b9-448c-9c73-29cd8a60d767","receiver_id":"120c100b-46c4-4481-b533-3e4f6a47b168","text":"libero non mattis pulvinar nulla pede ullamcorper augue a suscipit nulla elit ac nulla","created":"2005-08-30T04:31:56Z"}
```

Рисунок 6 – Запрос состояния и сущности

1.6 В задаче shadowJar добавить к jar-файлу вашу фамилию

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

Рисунок 7 – Добавление фамилии к jar-файлу при создании



 trpp-second-6-0.1	10.03.2024 22:12	Executable Jar File	20 965 КБ
 trpp-second-6-Potemkin-0.1	10.03.2024 22:35	Executable Jar File	20 965 КБ

Рисунок 8 – Результат shadowJar

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

```
> Task :checkstyleMain FAILED
[ant:checkstyle] [ERROR] C:\Users\lenovo\IdeaProjects\trpp-second-6\src\main\java\ru\mirea\trpp_second_6\Application.java:1:9: Name 'ru.mirea.trpp_second_6' must match pattern '^[a-z](\.[a-z])
[ant:checkstyle] [ERROR] C:\Users\lenovo\IdeaProjects\trpp-second-6\src\main\java\ru\mirea\trpp_second_6\controllers\HealthController.java:1:9: Name 'ru.mirea.trpp_second_6.controllers' must ma
]{1,})*$'. [PackageName]
[ant:checkstyle] [ERROR] C:\Users\lenovo\IdeaProjects\trpp-second-6\src\main\java\ru\mirea\trpp_second_6\controllers\MessageController.java:1:9: Name 'ru.mirea.trpp_second_6.controllers' must m
9){1,})*$'. [PackageName]
[ant:checkstyle] [ERROR] C:\Users\lenovo\IdeaProjects\trpp-second-6\src\main\java\ru\mirea\trpp_second_6\controllers\MessageController.java:21: Line is longer than 120 characters (found 170).
[ant:checkstyle] [ERROR] C:\Users\lenovo\IdeaProjects\trpp-second-6\src\main\java\ru\mirea\trpp_second_6\entity\HealthResponse.java:1:9: Name 'ru.mirea.trpp_second_6.entity' must match pattern
PackageName]
[ant:checkstyle] [ERROR] C:\Users\lenovo\IdeaProjects\trpp-second-6\src\main\java\ru\mirea\trpp_second_6\entity\Message.java:1:9: Name 'ru.mirea.trpp_second_6.entity' must match pattern '^[a-z]
Name]
FAILURE: Build failed with an exception.
```

Рисунок 9 – Выполнение checkstyleMain

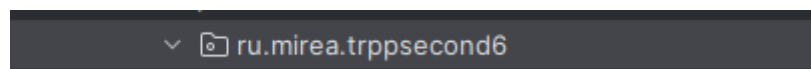


Рисунок 10 – Исправление ошибок

```
public MessageController() {
    messageList = new CsvToBeanBuilder<Message>(new InputStreamReader(this.getClass()
        .getResourceAsStream( name: "/MOCK_DATA.csv")))
        .withType(Message.class) CsvToBeanBuilder<Message>
        .build() CsvToBean<Message>
        .parse();
}
```

Рисунок 11 – Исправление ошибок

```
PS C:\Users\lenovo\IdeaProjects\trpp-second-6> ./gradlew checkstyleMain

> Task :compileJava
Note: Creating bean classes for 2 type elements
warning: A file for type 'ru.mirea.trppsecond6.controllers.$HealthControllerDefinition' already exists on the sourcepath or classpath
warning: A file for type 'ru.mirea.trppsecond6.controllers.$HealthControllerDefinition$exec1' already exists on the sourcepath or classpath
warning: A file for type 'ru.mirea.trppsecond6.controllers.$HealthControllerDefinitionClass' already exists on the sourcepath or classpath
warning: No processor claimed any of these annotations: /io.micronaut.http.annotation.Controller,/io.micronaut.http.annotation.Get
4 warnings

BUILD SUCCESSFUL in 3s
```

Рисунок 12 – Выполнение checkstyleMain

2 Ответы на вопросы

8. Что делает задача clean?

Задача clean в Gradle удаляет каталог сборки проекта, очищая все скомпилированные классы и другие результаты сборки.

9. Что делает задача build?

Задача build в Gradle является агрегирующей задачей, которая выполняет полный цикл сборки проекта. Она компилирует и тестирует ваш код, а затем собирает его в исполняемый jar-файл или war-файл (для веб-приложений).

10. Что делает задача compileJava?

Задача compileJava в Gradle компилирует исходный код Java проекта. Она преобразует файлы .java в байт-код Java, который сохраняется в файлах .class.

11. Что делает задача run?

Задача run в Gradle запускает приложение, используя определенную в проекте точку входа. Обычно это используется для запуска приложений, написанных на языках, таких как Java или Groovy.

12. Что делает задача test?

Задача test в Gradle запускает автоматические тесты, которые были написаны для приложения. Это могут быть модульные тесты, интеграционные тесты или любой другой тип тестов, которые вы написали.

13. Что такое javadoc?

Javadoc - это инструмент, включенный в JDK (Java Development Kit), который генерирует документацию API для исходного кода Java. Он читает комментарии Javadoc в исходном коде и создает HTML-

страницы, которые описывают публичные и защищенные классы, интерфейсы, конструкторы, методы и поля каждого пакета.

Вывод

В ходе выполнения этой практической работы мы углубили свои знания в области использования системы сборки Gradle. Мы изучили основные задачи, такие как `clean`, `build`, `checkstyleMain`, `shadowJar`, `run`, `test` и т.д. Также научились управлять зависимостями в проекте.