



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«МИРЭА - Российский технологический университет»

**РТУ МИРЭА**

---

**Институт информационных технологий (ИИТ)**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**  
по дисциплине «Технология разработки программных приложений»

**Практическое задание № 3**

Студент группы

ИКБО-04-22 Арефьев А.М

\_\_\_\_\_  
(подпись)

Отчет представлен

«\_\_» \_\_\_\_\_ 2024г.

## Оглавление

ЗАДАНИЕ.....	3
Цель работы:.....	3
Вариант.....	3
Задание.....	3
Вопросы.....	7
Вывод.....	8

# ЗАДАНИЕ

## Цель работы:

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

## Вариант

1) репозиторий: <https://github.com/rtu-mirea/trpp-second-1>, сущность ru.mirea.entity.Employee

## Задание

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.

1. Найденные зависимости: opencsv

В файл build.gradle (dependencies) добавлено:

```
implementation("com.opencsv:opencsv:4.0")
```

2. Также убираем нижние подчеркивания в названии пакетов и `import com.opencsv.bean.CsvToBeanBuilder` в файл `EmployeeController`.

3. Теперь проект запускается, но появляется ошибка стиля, открываем `html` и исправляем.

CheckStyle Audit

Designed for use with [CheckStyle](#) and [Ant](#).

Summary	
Files	Errors
5	3

Files	
Name	Errors
<a href="#">/home/merch/repos/tpsp-second-1/src/main/java/ru/mirea/tpsecond1/controllers/EmployeeController.java</a>	3
<a href="#">/home/merch/repos/tpsp-second-1/src/main/java/ru/mirea/tpsecond1/Application.java</a>	0
<a href="#">/home/merch/repos/tpsp-second-1/src/main/java/ru/mirea/tpsecond1/controllers/HealthController.java</a>	0
<a href="#">/home/merch/repos/tpsp-second-1/src/main/java/ru/mirea/tpsecond1/entity/Employee.java</a>	0
<a href="#">/home/merch/repos/tpsp-second-1/src/main/java/ru/mirea/tpsecond1/entity/HealthResponse.java</a>	0

File /home/merch/repos/tpsp-second-1/src/main/java/ru/mirea/tpsecond1/Application.java	
Error Description	Line
<a href="#">Back to top</a>	

File /home/merch/repos/tpsp-second-1/src/main/java/ru/mirea/tpsecond1/controllers/EmployeeController.java	
Error Description	Line
Wrong order for 'io.micronaut.http.HttpResponse' import.	4
Wrong order for 'com.opencsv.bean.CsvToBeanBuilder' import.	7
Line is longer than 120 characters (found 173).	21
<a href="#">Back to top</a>	

File /home/merch/repos/tpsp-second-1/src/main/java/ru/mirea/tpsecond1/controllers/HealthController.java	
Error Description	Line
<a href="#">Back to top</a>	

File /home/merch/repos/tpsp-second-1/src/main/java/ru/mirea/tpsecond1/entity/Employee.java	
Error Description	Line
<a href="#">Back to top</a>	

File /home/merch/repos/tpsp-second-1/src/main/java/ru/mirea/tpsecond1/entity/HealthResponse.java	
Error Description	Line
<a href="#">Back to top</a>	

Рисунок 1 — checkstyle

Исправляем ошибку неправильного расположения импортов.

```
1 package ru.mirea.trppsecond1.controllers;
2
3 import com.opencsv.bean.CsvToBeanBuilder;
4 import io.micronaut.http.HttpResponse;
5 import io.micronaut.http.annotation.Controller;
6 import io.micronaut.http.annotation.Get;
7 import ru.mirea.trppsecond1.entity.Employee;
8 import java.io.InputStreamReader;
9 import java.util.List;
10 import java.util.Optional;
```

## Рисунок 2 — Правильное расположение импортов

#### 4. Собираем jar и запускаем сервер

```
~/repos/trpp-second-1 master +10 13 ./gradlew build
> Task :compileJava
Note: Creating bean classes for 1 type elements
warning: No processor claimed any of these annotations: /io.micronaut.http.annotation.Controller,/io.micronaut.http.annotation.Get
1 warning

BUILD SUCCESSFUL in 5s
13 actionable tasks: 12 executed, 1 up-to-date

~/repos/trpp-second-1 master +10 13 ./gradlew shadowjar
Starting a Gradle Daemon, 2 stopped Daemons could not be reused, use --status for details

BUILD SUCCESSFUL in 4s
3 actionable tasks: 3 up-to-date
```

### Рисунок 3 — Успешный запуск

[illegible]

### Рисунок 3.1 — Успешный запуск

## 5. Запросим состояние.

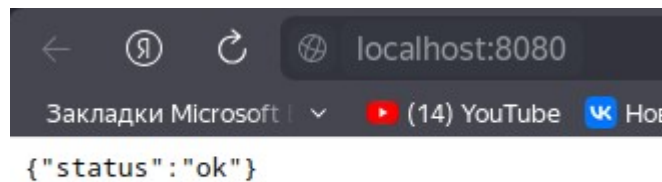


Рисунок 4 — Состояние сервера

## 6. Запросим сущность.

```
{"id":131,"email":"jgallety3m@imdb.com","firstName":"Jock","lastName":"Gallety","birthDate":"2008-07-28T20:51:02Z","company":"Streich, Ratke and Dooley","department":"Product Management"}
```

Рисунок 5 — Запрос сущности

## 7. Добавим в shadowjar мою фамилию.

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

Рисунок 6 — shadowjar

## 8. Финальная проверка checkstyle

```
~/repos/trpp-second-1 master +10 13 ./gradlew checkstyleMain
BUILD SUCCESSFUL in 713ms
3 actionable tasks: 3 up-to-date
```

Рисунок 7 — Checkstyle

# ВОПРОСЫ

**1) Чем компиляция отличается от сборки?** Компиляция отличается от сборки тем, что компиляция преобразует исходный код программы в машинный код или другой исполняемый формат, в то время как сборка включает в себя этот процесс компиляции, а также другие этапы.

**2) Что такое система сборки?** Система сборки представляет собой инструмент или набор инструментов, который автоматизирует процесс компиляции, тестирования и развертывания программного обеспечения.

**3) Что такое репозиторий?** Репозиторий — это централизованное хранилище для хранения и управления версиями файлов или данных.

**4) Как указать зависимости проекта?** Зависимости проекта могут быть указаны в специальных файлах (например, `build.gradle` для Gradle или `pom.xml` для Maven) с помощью специального синтаксиса.

**5) Что такое Gradle?** Gradle — это инструмент с открытым исходным кодом для сборки проектов.

**6) Что делает задача `build`?** Задача `build` компилирует и собирает проект, создавая исполняемые файлы или артефакты.

## **ВЫВОД**

Познакомился с системой сборки gradle, ее возможностями, еще больше убедился в том, что работать с джавой мне не хочется.