|  |
| --- |
|  |
| 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 № 2** |
| **по дисциплине**  «ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ» |
| **Тема: «**Системы сборки**»** |

|  |  |
| --- | --- |
| Выполнил студент группы ИКБО-32-21 | Быченков А.К. |
| Принял | Миронов А.И. |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | «\_\_»\_\_\_\_\_\_\_2023 г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2023 г. | *(подпись руководителя)* |

Москва 2023

**Задание**

**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 № 4**: репозиторий: <https://github.com/rtu-mirea/trpp-second-4>, сущность ru.mirea.entity.Organization.

**Ход работы**

1. Для корректной сборки проекта была добавлена отсутствующая зависимость com.opencsv:opencsv:

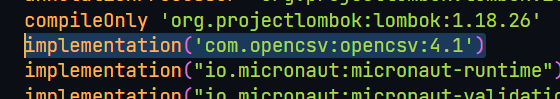


Рисунок 1. Добавленная зависимость

1. Для mainClass было поправлено имя пакета:



Рисунок 2. Исправление имени пакета

1. Командой gradle javadoc была создана документация проекта. В ней были найдены запросы состояния и сущности по идентификатору:

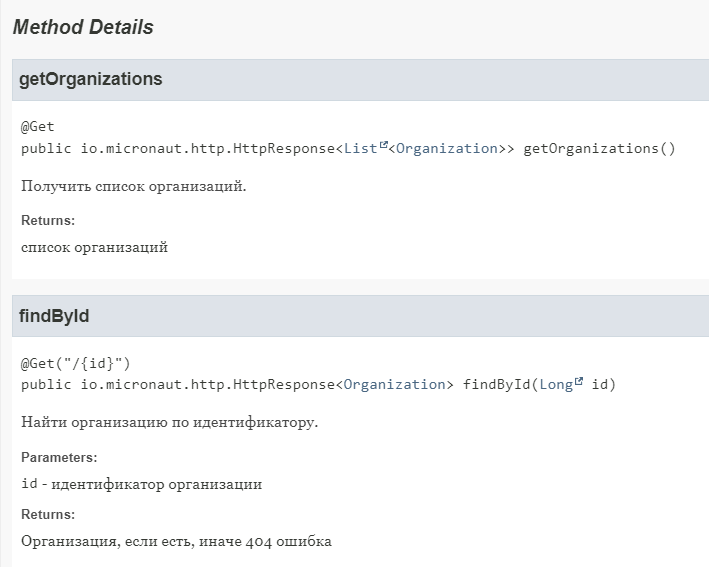


Рисунок 3. Документация по запросам

1. Командой gradle shadowJar был создан UberJar. После чего приложение было запущено:

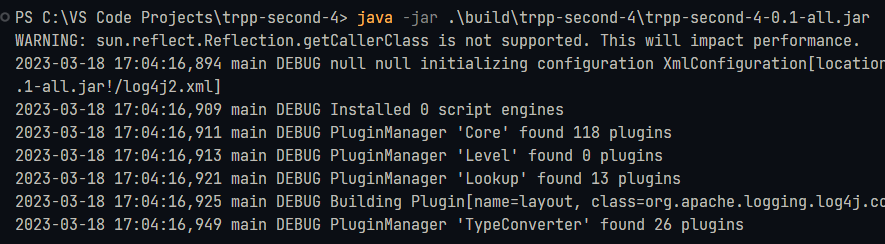


Рисунок 4. Запуск приложения

1. Был отправлен GET-запрос на сервер по адресу <http://localhost:8080> и получен ответ:

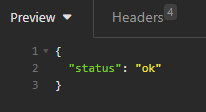


Рисунок 5. Полученный ответ

1. Был отправлен GET-запрос на сервер по адресу <http://localhost:8080/organization/483> и получен ответ в виде сущности организации с идентификатором 483:

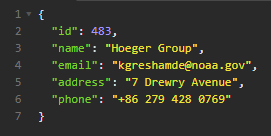


Рисунок 6. Полученный ответ

1. В задаче shadowJar к имени jar-файла была добавлена фамилия:

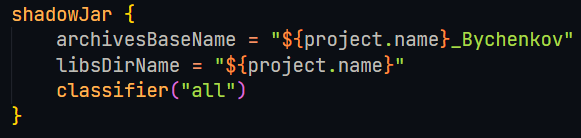


Рисунок 7. Задача shadowJar

1. Была выполнена задача checkstyleMain:

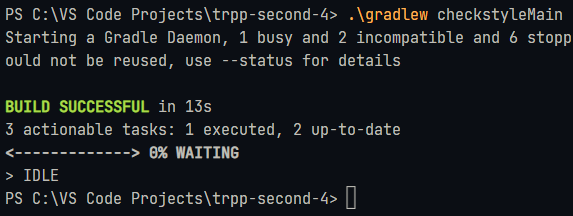


Рисунок 8. Запуск задачи checkstyleMain

Все ошибки оформления кода были исправлены:

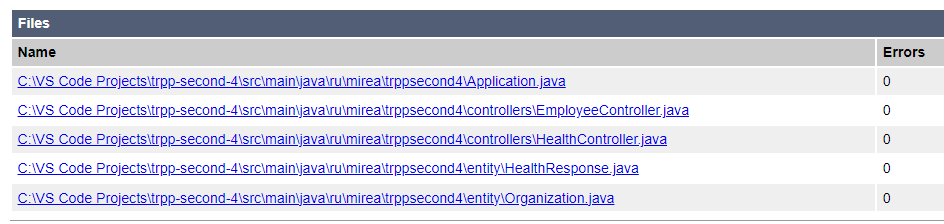


Рисунок 9. Отчет об ошибках оформления кода

**Ответы на контрольные вопросы**

1. **Что делает задача run?**

Задача run в Gradle запускает приложение, которое было предварительно скомпилировано. Если приложение не было скомпилировано, то перед запуском будет выполнена задача compileJava для компиляции исходного кода приложения.

1. **Что такое UberJar? При помощи какой задачи его собрать?**

UberJar — это один большой исполняемый файл, содержащий все зависимости приложения. Это удобно для развертывания приложения на удаленной машине, где может отсутствовать подключение к Интернету для загрузки зависимостей в процессе работы. Задача shadow в Gradle используется для создания UberJar.

1. **Что такое javadoc?**

javadoc – это утилита для генерации документации на основе комментариев в исходном коде на языке Java. Это позволяет создавать понятную документацию для разработчиков, использующих ваш код.

1. **Чем компиляция отличается от сборки?**

Компиляция – это процесс преобразования исходного кода на языке программирования в байт-код, который может быть выполнен на виртуальной машине Java. Сборка — это процесс создания исполняемого файла или библиотеки на основе скомпилированного кода и всех его зависимостей.

1. **Что такое lombok?**

lombok – это библиотека для языка Java, которая упрощает написание кода, добавляя аннотации для автоматической генерации геттеров, сеттеров, конструкторов и других методов, которые часто используются в Java.

1. **Что такое аннотация в Java?**

Аннотация в Java – это механизм, который позволяет добавлять метаданные к коду. Аннотации используются для описания дополнительной информации, которая может использоваться компилятором, средой разработки или другими инструментами во время выполнения приложения. Аннотации могут быть использованы для определения конфигурации, обработки ошибок, описания API и т.д.

**Вывод**

В результате выполнения работы я познакомился с системой сборки Gradle. Изучил её основные возможности и управление зависимостями.