МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ **«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА» (БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №3  
по дисциплине: Технологии Web-программирования

Выполнил: Ляхов Степан ПВ-42

Проверил: Картамышев С. В.

Белгород 2020г.

**Цель**: познакомиться с основами backend разработки web-приложений. Научится писать программы на языке Java. Познакомиться с основами работы docker. Познакомиться с фреймворком Spring Boot и научиться разворачивать проект, производить его настройку. Научится работать с API в приложении Postman.

**Задание к лабораторной работе:**

1. Развернуть базовое приложение Spring Boot App Basic.
2. Настроить конфигурацию работы приложения с docker.
3. Добавить модуль для работы с API.
4. Добавить несколько контроллеров со статическими данными.
5. Продемонстрировать работу API в Postman.

**Выполнение**

В данной лабораторной работе мы будем реализовывать простой Rest сервис, с двумя endpoints.

Язык программирования: Java 8

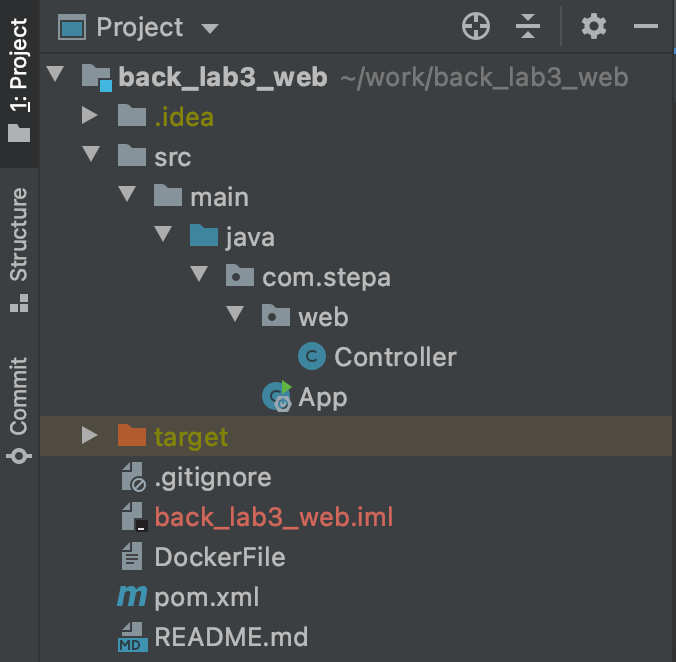
Фреймворк: Spring, Spring Boot

Система сборки: Maven

Контейниризатор: Docker

После установки java8 и maven

Создаём новый проект с такой структурой:



pom.xml – файл в котором будут находится все необходимые нам зависимости

DockerFile – файл, в котом будет находится конфигурация нашего Docker-образа

README.md – файл, в котором будет находиться инструкция по использованию.

App – класс, точка запуска приложения (настройка конфигурации и многое другое)

Controller – класс, контроллер. Реализация endpoints

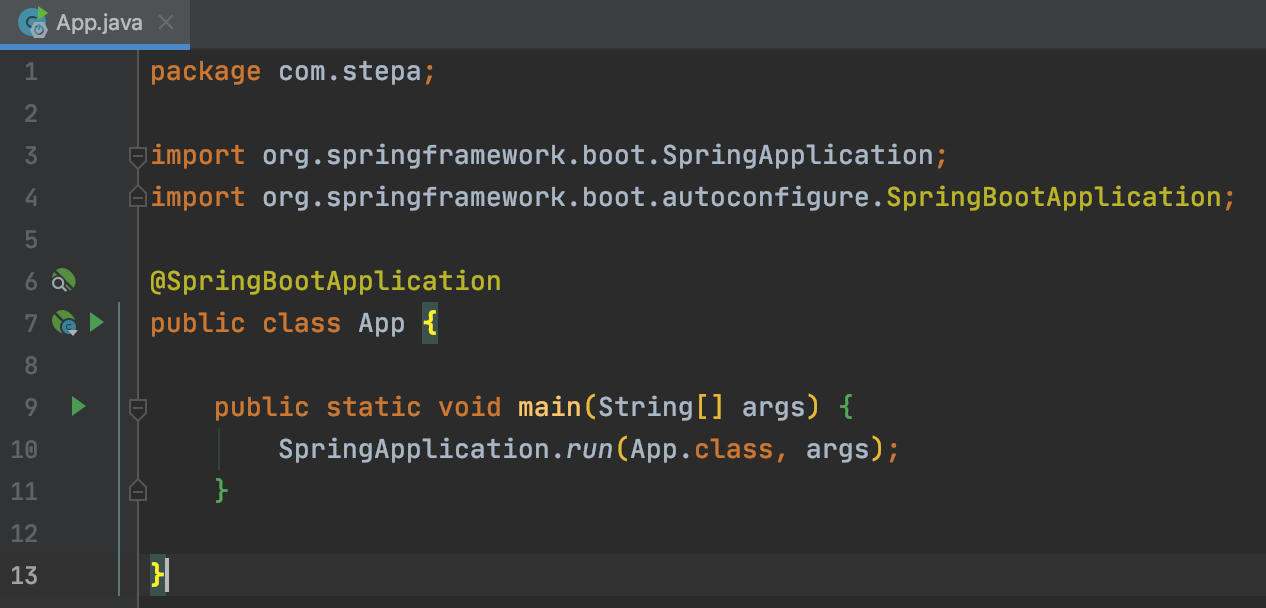
**Подключение необходимых библиотека**

Нам понадобиться библиотека для реализации контролеров, поднятия tomcat.



Maven автоматические библиотеки для работы текущей, такие как Spring DI для внедрения зависимостей.

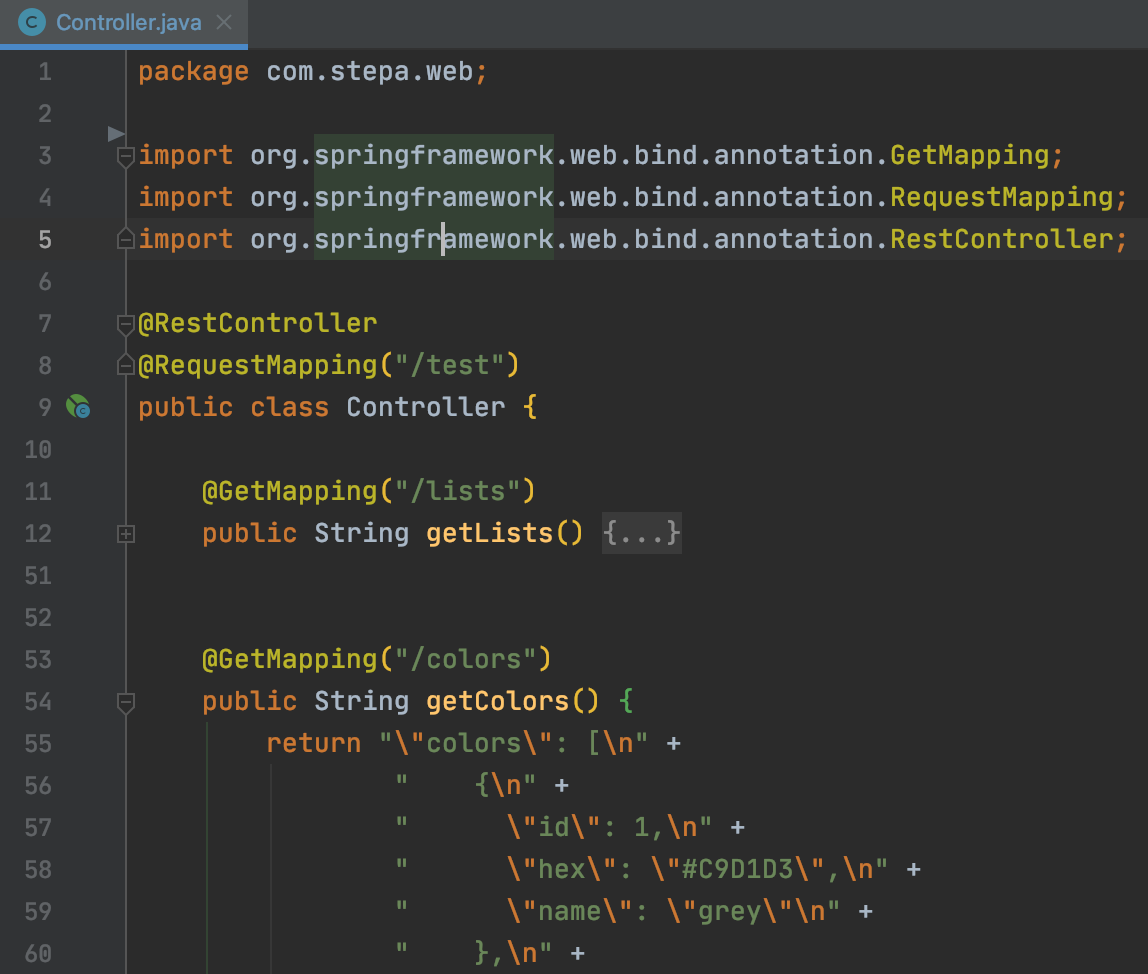
**Создаём точку запуска нашего приложения**

****

Так выглядит наш базовый класс. Тут ним интересно две вещи:

@SpringBootApplication – аннотация фреймворка, которая делает очень кучу всего. Но нам важно, что она возьмёт свою базовую конфигурацию и настроит зависимости, заинжектит, запустит tomcat по дефолту на 8080 порту.

**Создаём контроллеры**



@RestController – аннотация, которая говорит, что этот класс является компонентом и так же является REST контроллером

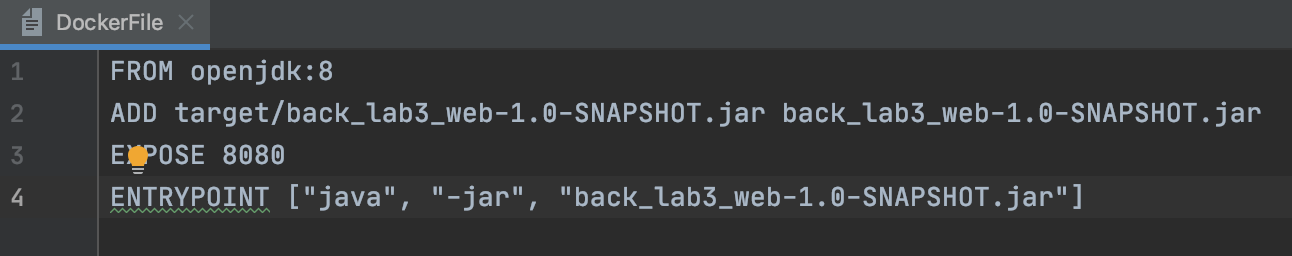
@RequestMapping("/test") – url этого контроллера начнётся с /test

Над методами находится аннотация @GetMapping("/lists") – которая говорит, какой будет тип запроса (GET, POST, DEL и.т.д.) и так же часть его url.

В нашем случае методы будут отдавать json в виде строки.

**Добавление приложения в Docker**

После установки Docker на компьютер, мы должны прописать в docker-файл, что нужно сделать.



Скачать образ с java8.

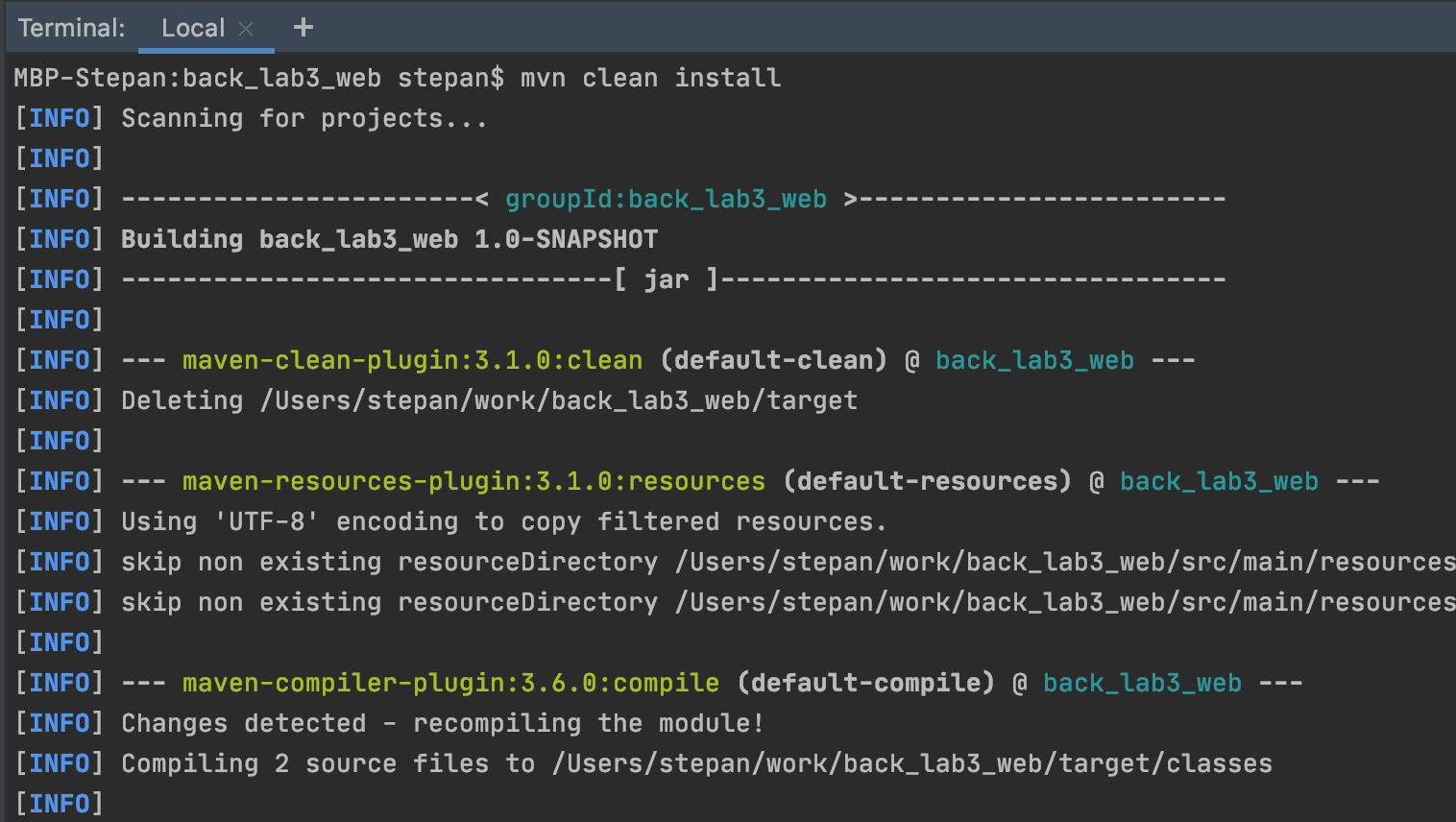
Скопировать скомпилированное приложение в образ.

Выдать порт.

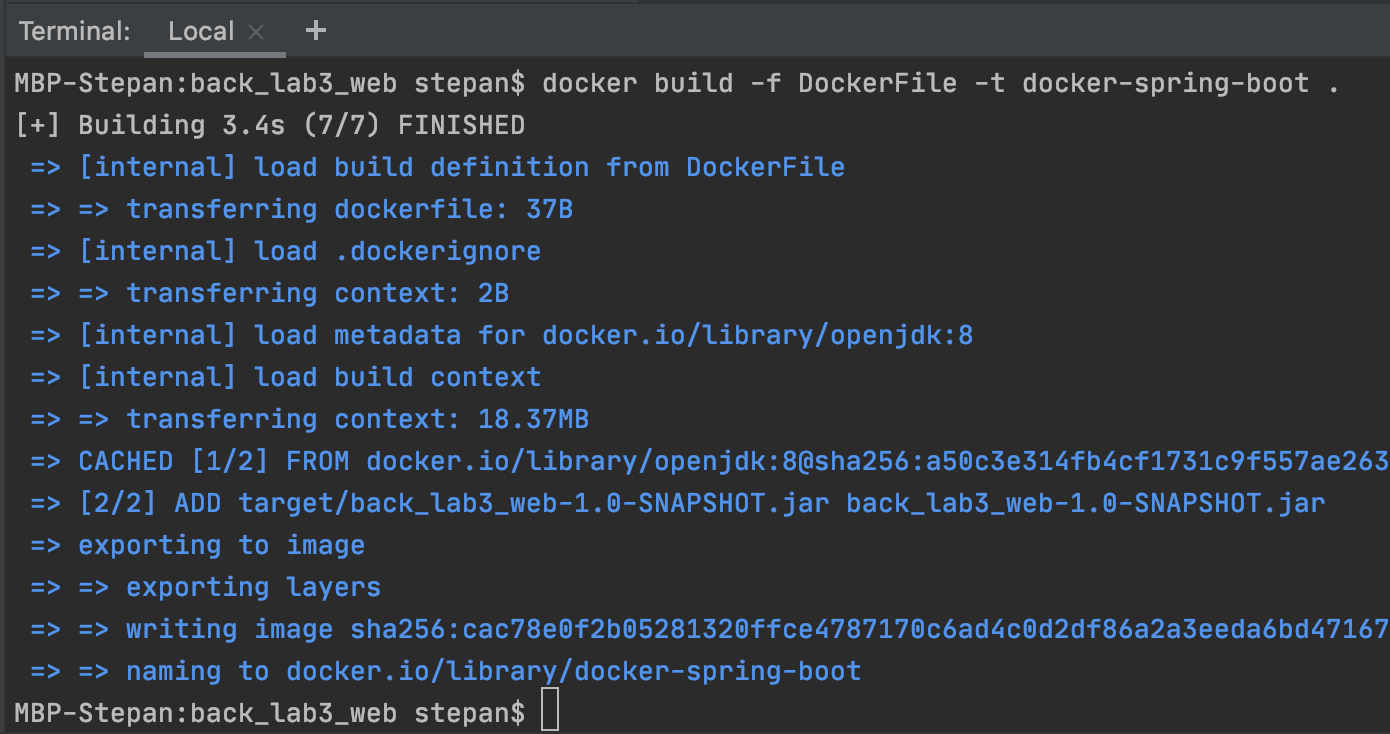
Запустить команду для запуска приложения.

**Создание и запуск Docker-образа**

Компилируем приложение и создаём из него jar



Собираем образ



Запускам приложение

