Домашняя работа

## по курсу “Фреймворк Spring”

Выполнил: Байбородин Александр

Группа: Программист | 17.04.2024

### Урок 1. Системы сборки Maven и Gradle для разработки Java приложений

Задание

Задание: Создайте простое веб-приложение с использованием Maven и Gradle. Сравните их возможности и поделитесь своими мыслями.

Решение:

Сформируем проект maven командой:

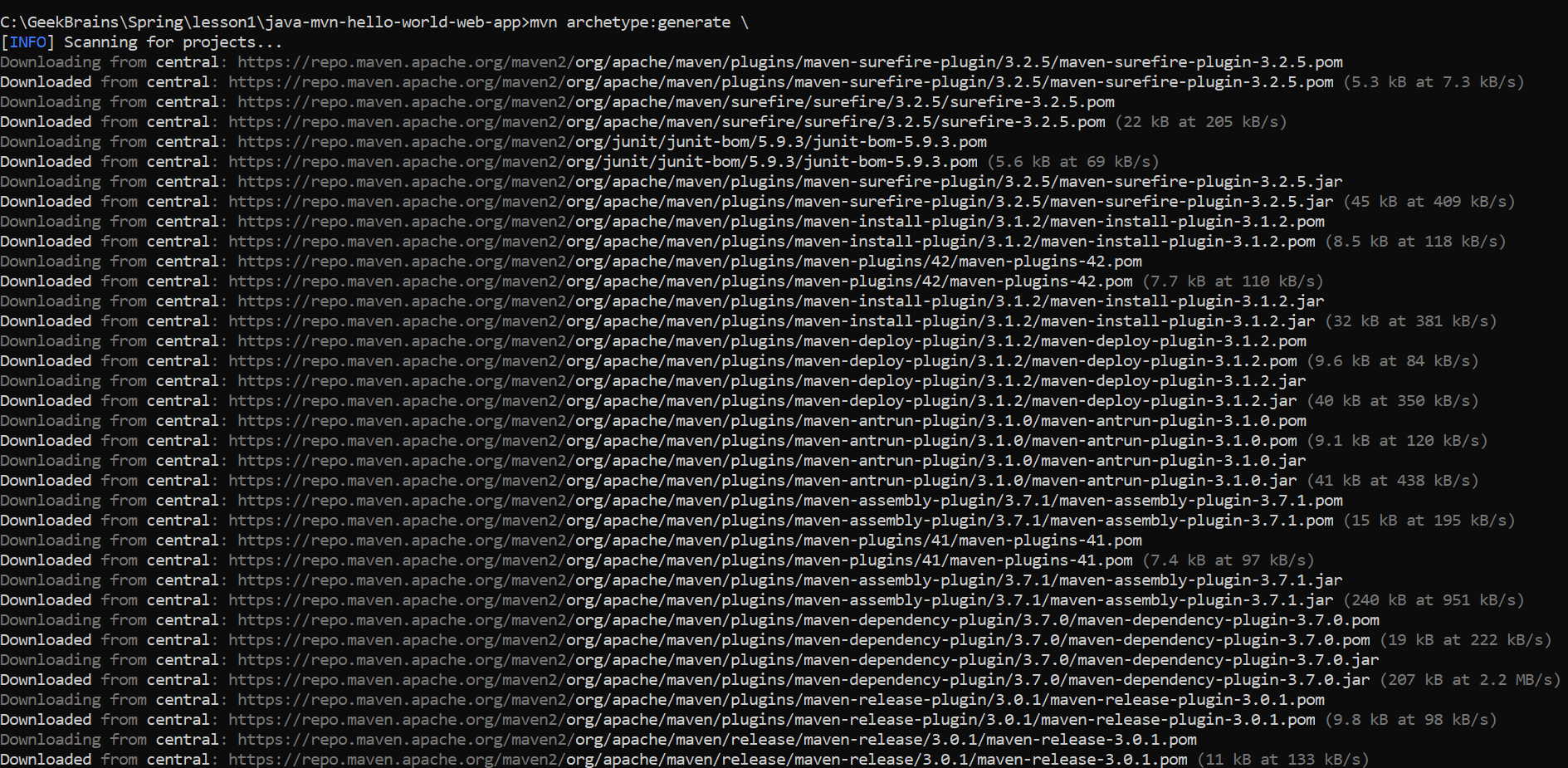
mvn archetype:generate \

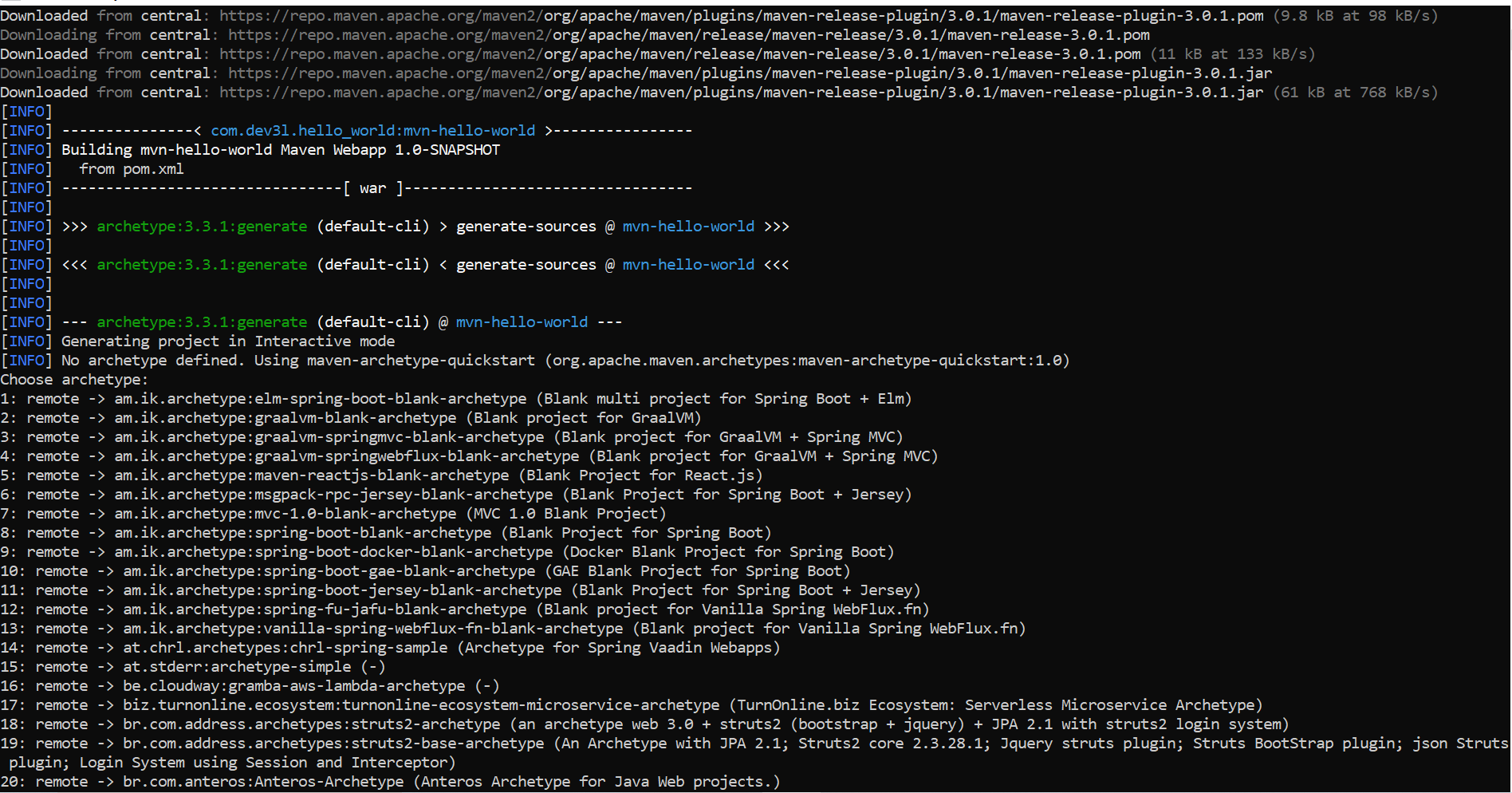
-DgroupId=com.example \

-DartifactId=simple-web-app \

-DarchetypeArtifactId=maven-archetype-webapp \

-DinteractiveMode=false





Добавим в pom.xml-файл необходимые для формирования web-приложения зависимости:

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>com.dev3l.hello\_world</groupId>  
 <artifactId>mvn-hello-world</artifactId>  
 <packaging>war</packaging>  
 <version>1.0-SNAPSHOT</version>  
 <name>mvn-hello-world Maven Webapp</name>  
 <url>http://maven.apache.org</url>  
 <dependencies>  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>servlet-api</artifactId>  
 <version>2.5</version>  
 <scope>provided</scope>  
 </dependency>  
 <!-- junit -->  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>4.12</version>  
 <scope>test</scope>  
 </dependency>  
 <!-- log4j -->  
 <dependency>  
 <groupId>org.apache.logging.log4j</groupId>  
 <artifactId>log4j-api</artifactId>  
 <version>2.5</version>  
 </dependency>  
 <dependency>  
 <groupId>org.apache.logging.log4j</groupId>  
 <artifactId>log4j-core</artifactId>  
 <version>2.5</version>  
 </dependency>  
 </dependencies>  
 <build>  
 <finalName>mvn-hello-world</finalName>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.1</version>  
 <configuration>  
 <source>1.8</source>  
 <target>1.8</target>  
 </configuration>  
 </plugin>  
 <plugin>  
 <!-- Java EE 6 doesn't require web.xml, Maven needs to catch up! -->  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-war-plugin</artifactId>  
 <version>2.4</version>  
 <configuration>  
 <failOnMissingWebXml>false</failOnMissingWebXml>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
</project>

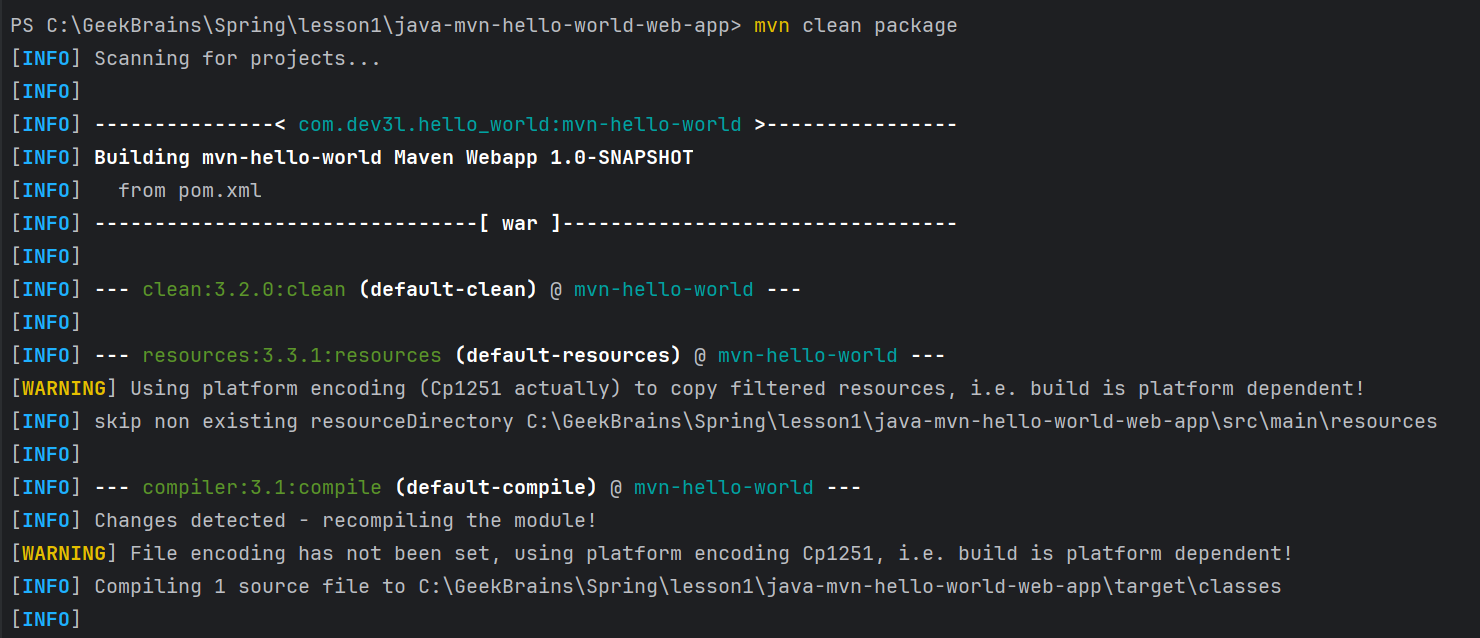
Создадим сервлет веб-приложения java/HelloServlet.java:

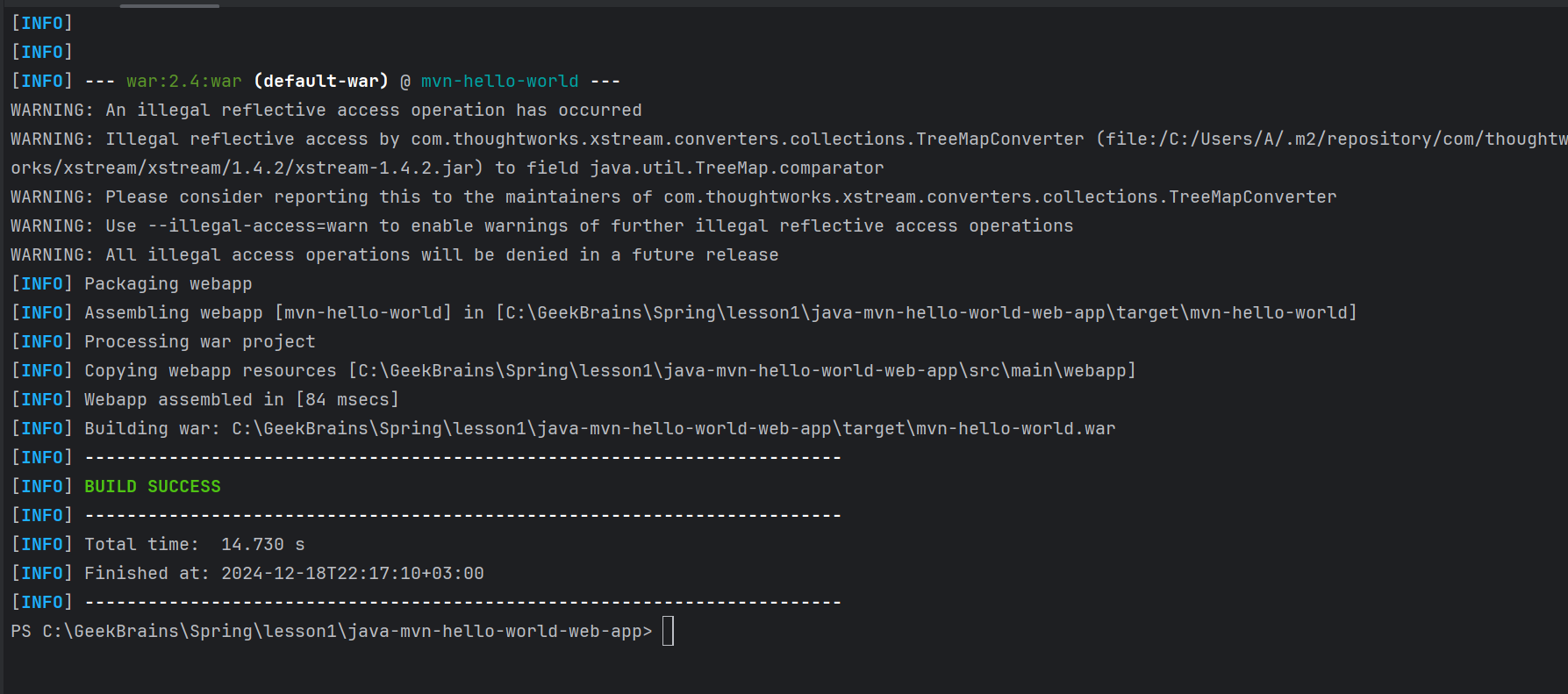
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
  
public class HelloServlet extends HttpServlet {  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 resp.getWriter().println("Hello, World!");  
 }  
}

Отредактируем файл web.xml, который находится в папке src/main/webapp/WEB-INF, чтобы зарегистрировать сервлет и сопоставить его с URL:

<?xml version="1.0" encoding="UTF-8"?>  
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app\_2\_4.xsd"  
 version="2.4">  
  
 <display-name>Simple Web App</display-name>  
  
 <servlet>  
 <servlet-name>helloServlet</servlet-name>  
 <servlet-class>com.example.HelloServlet</servlet-class>  
 </servlet>  
  
 <servlet-mapping>  
 <servlet-name>helloServlet</servlet-name>  
 <url-pattern>/hello</url-pattern>  
 </servlet-mapping>  
  
</web-app>

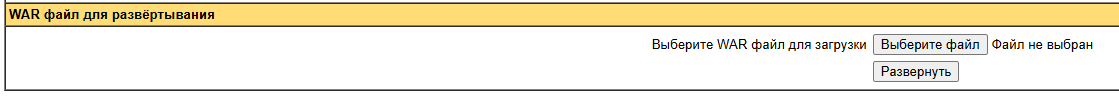
Для сборки проекта выполним следующую команду:

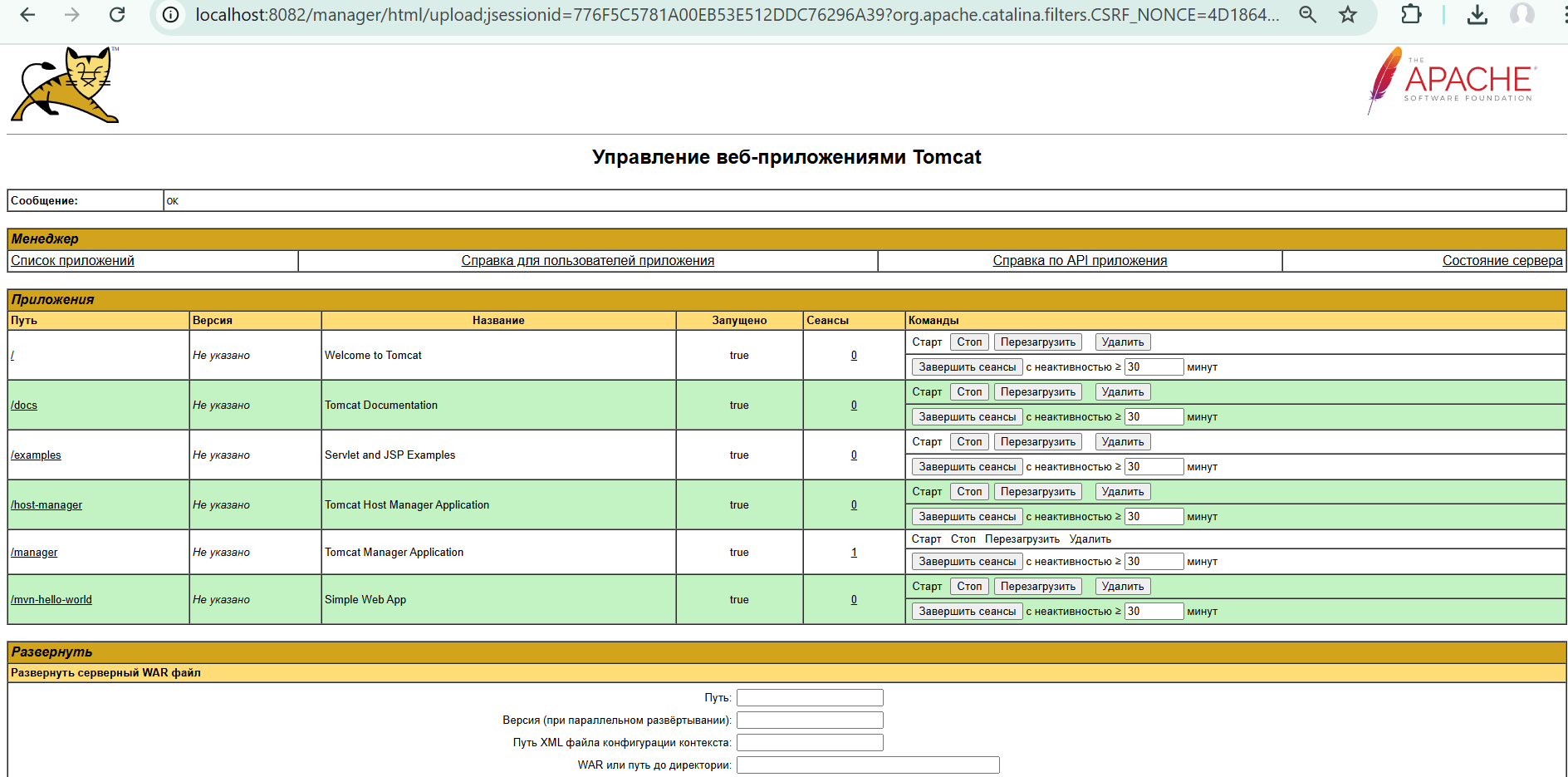


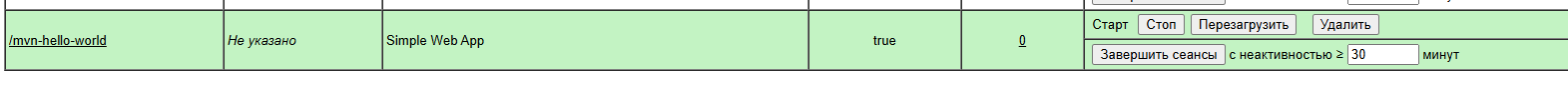


Сборка проекта создаст WAR-файл (simple-web-app.war) в папке target. Теперь мы можем развернуть этот WAR-файл на сервере приложений, таком как Tomcat.

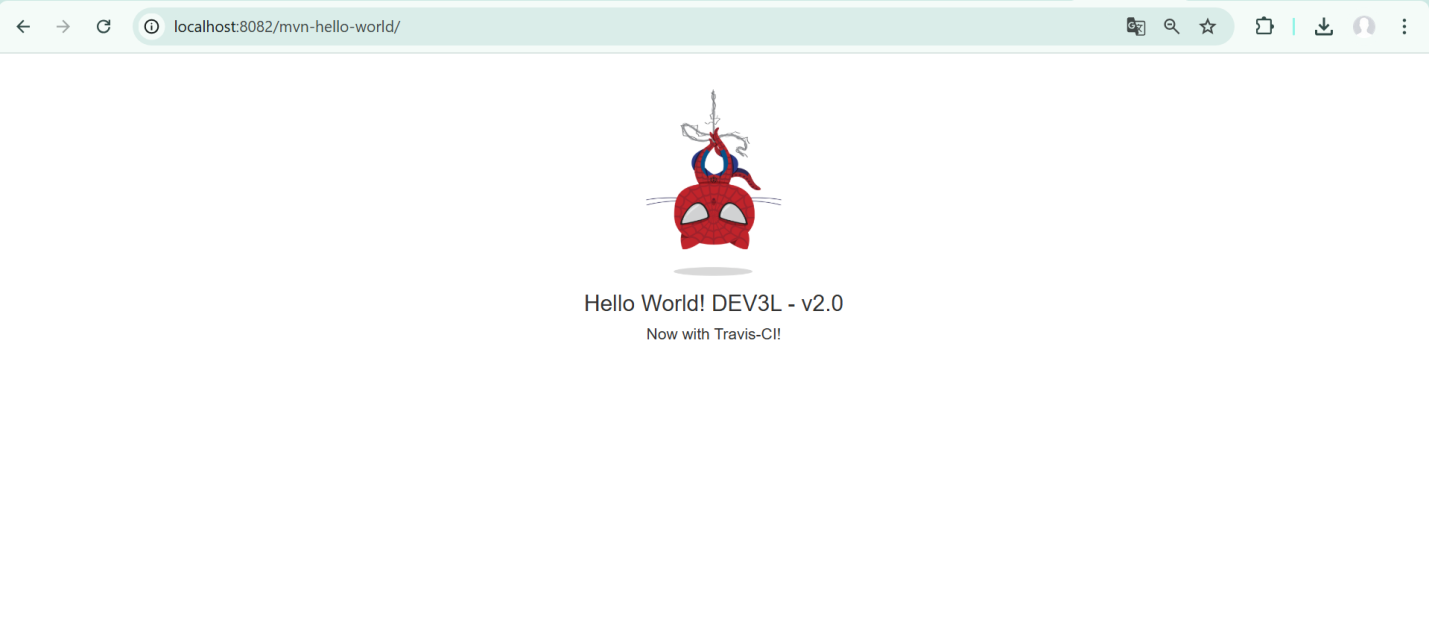
Развернем war-файл на сервере Tomkat, выбрав его:



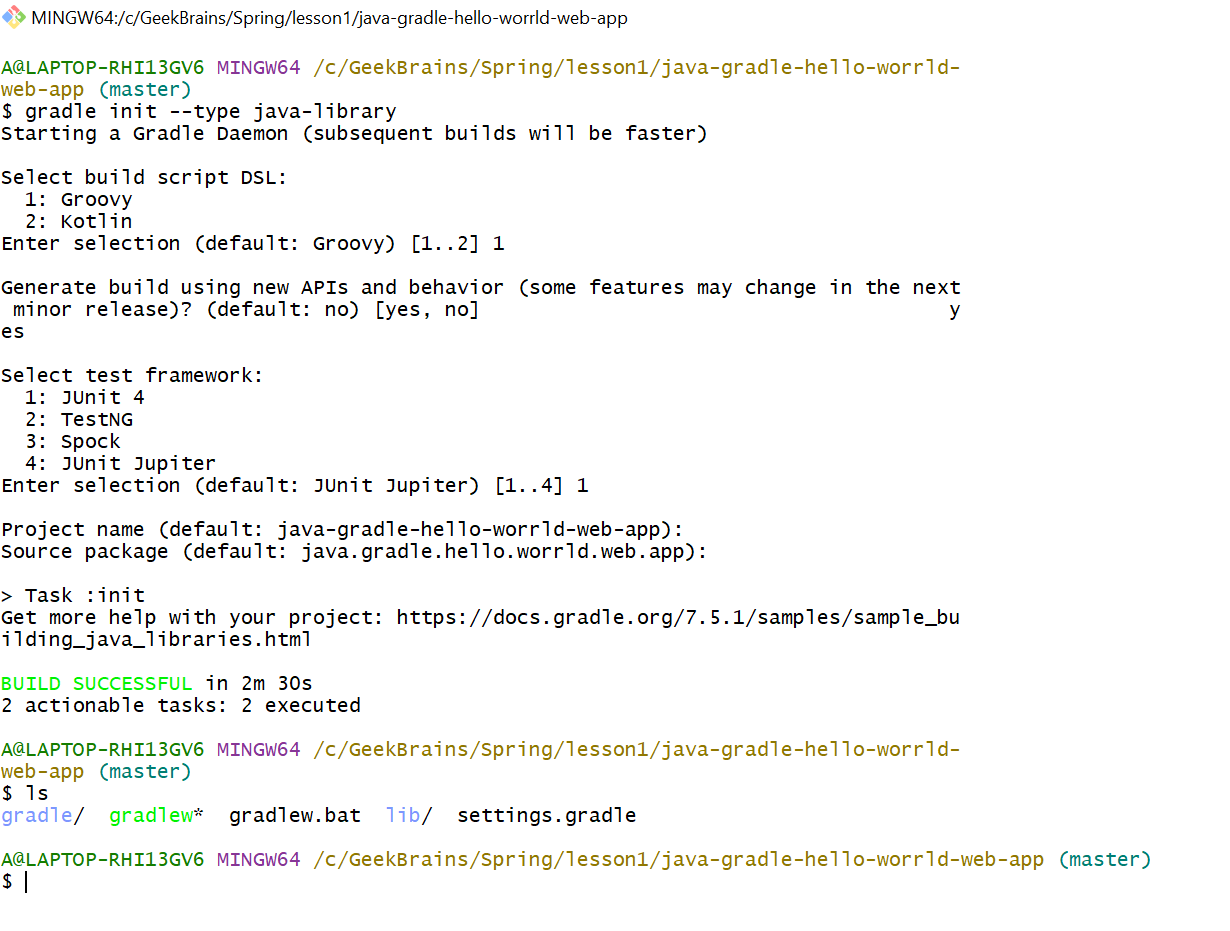




Посмотрим запущенный сайт:



Теперь создадим gradle-проект командой:

ОJОО

Отредактируем файл настроек build-gradle:

plugins **{** id 'war'  
 id 'java'  
**}**group = 'org.example'  
version = '1.0-SNAPSHOT'  
  
repositories **{** mavenCentral()  
**}**dependencies **{** testImplementation platform('org.junit:junit-bom:5.9.1')  
 testImplementation 'org.junit.jupiter:junit-jupiter'  
 implementation 'javax.servlet:javax.servlet-api:4.0.1'  
 implementation 'jakarta.servlet:jakarta.servlet-api:5.0.0'  
**}**test **{** useJUnitPlatform()  
**}**war **{** archiveBaseName = 'simple-web-app'  
 archiveVersion = '1.0'  
**}**

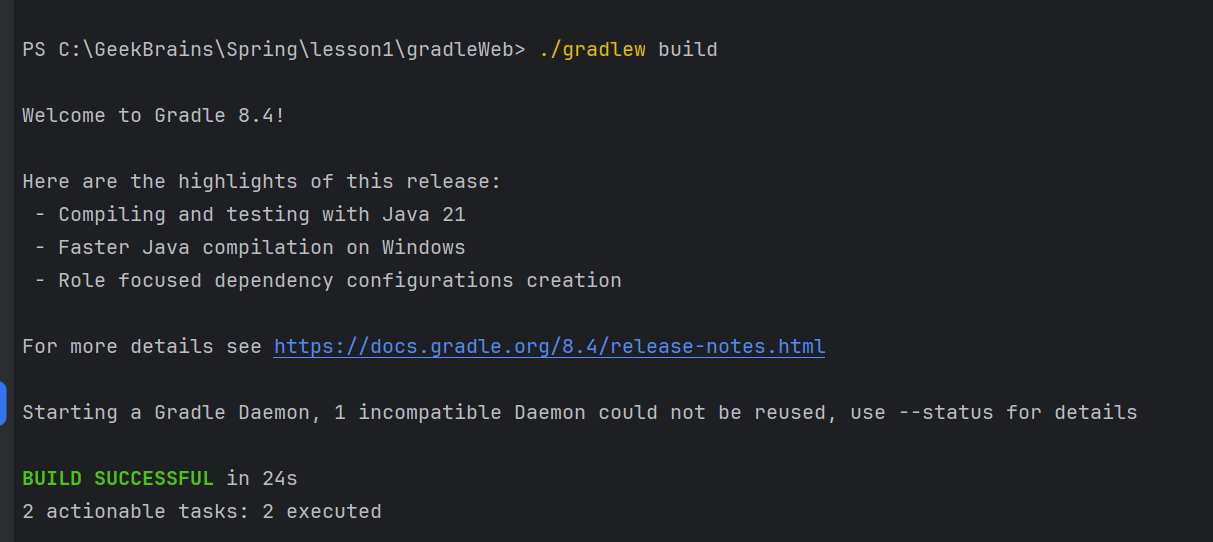
Создадим файл HelloServlet.java:

package org.example;  
  
import jakarta.servlet.ServletException;  
import jakarta.servlet.annotation.WebServlet;  
import jakarta.servlet.http.HttpServlet;  
import jakarta.servlet.http.HttpServletRequest;  
import jakarta.servlet.http.HttpServletResponse;  
import java.io.IOException;  
  
@WebServlet("/hello")  
public class HelloServlet extends HttpServlet {  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 resp.getWriter().println("Hello, World!");  
 }  
}

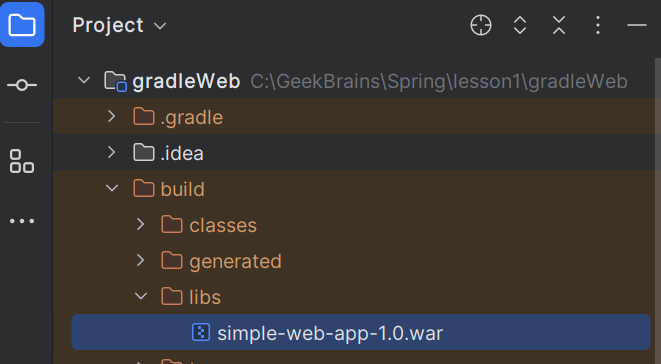
Создадим файл web.xml:

<?xml version="1.0" encoding="UTF-8"?>  
<web-app xmlns="https://jakarta.ee/xml/ns/jakartaee"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee https://jakarta.ee/xml/ns/jakartaee/web-app\_5\_0.xsd"  
 version="5.0">  
  
 <display-name>Simple Web App</display-name>  
  
 <servlet>  
 <servlet-name>helloServlet</servlet-name>  
 <servlet-class>com.example.HelloServlet</servlet-class>  
 </servlet>  
  
 <servlet-mapping>  
 <servlet-name>helloServlet</servlet-name>  
 <url-pattern>/hello</url-pattern>  
 </servlet-mapping>  
  
</web-app>

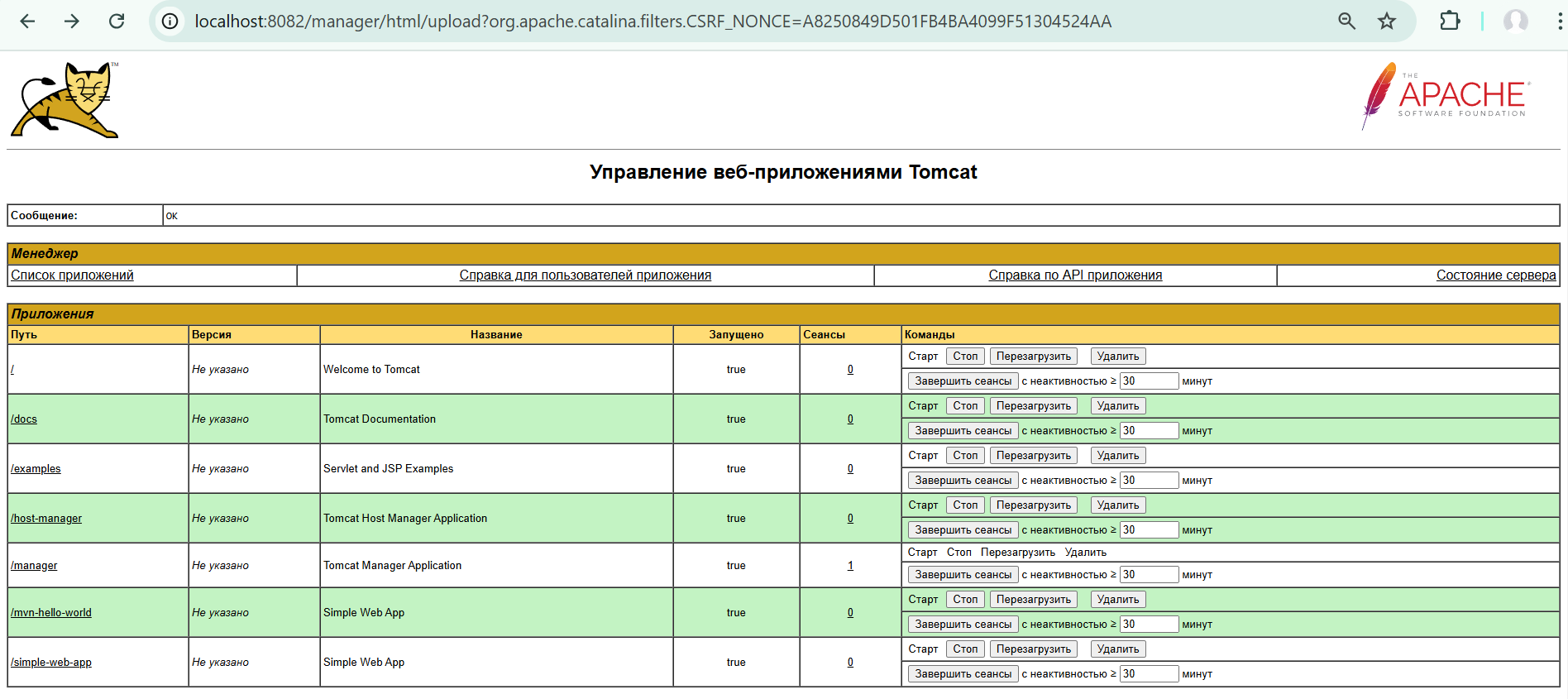
Запустим команду сборки проекта:



Сборка создаст WAR-файл (simple-web-app-1.0.war) в папке build/libs. Теперь мы можем развернуть этот WAR-файл на сервере приложений, таком как Tomcat. Откроем и развернем в Tomcat war-файл:







Откроем в браузере веб-приложение:

