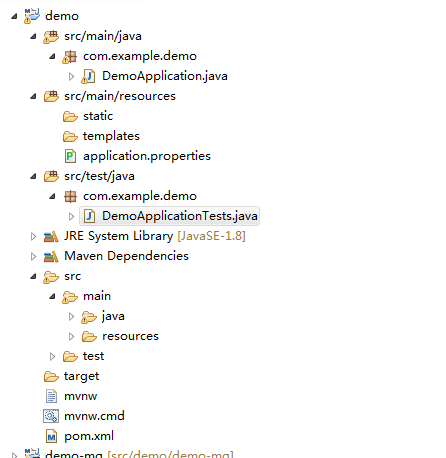
# 基础概念

1. SpringBoot核心
2. 自动配置：针对很多Spring应用程序常见的应用功能，Spring Boot能自动提供相关配置
3. 起步依赖：告诉Spring Boot需要什么功能，它就能引入需要的库。
4. 命令行界面：这是Spring Boot的可选特性，借此你只需写代码就能完成完整的应用程序，无需传统项目构建。
5. Actuator：让你能够深入运行中的Spring Boot应用程序，一探究竟。
6. 创建一个Spring Boot工程的方式
7. 下载Spring Initializr（<http://start.spring.io/）Maven>项目搭建工具
8. 利用eclipse导入springboot工程demo
9. 工程目录



1. 配置文件：pom.xml(Maven配置)，application.properties(全局变量配置文件，可以放空)
2. pom.xml需要导入的包以及插件

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

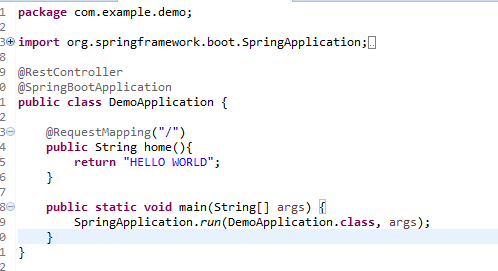
<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

1. 启动SpringBoot工程的方式
2. 直接通过main方法启动



1. 通过maven命令启动（略）
2. 打包工程成jar包，运行jar包启动（略）
3. 访问工程的方式

地址：localhost:8080/（Controller的名称）

# 依赖包介绍

spring-boot-starter

这是Spring Boot的核心启动器，包含了自动配置、日志和YAML。

spring-boot-starter-amqp

通过spring-rabbit来支持AMQP协议（Advanced Message Queuing Protocol. 。

spring-boot-starter-aop

支持面向方面的编程即AOP，包括spring-aop和AspectJ。

spring-boot-starter-artemis

通过Apache Artemis支持JMS的API（Java Message Service API. 。

spring-boot-starter-batch

支持Spring Batch，包括HSQLDB数据库。

spring-boot-starter-cache

支持Spring的Cache抽象。

spring-boot-starter-cloud-connectors

支持Spring Cloud Connectors，简化了在像Cloud Foundry或Heroku这样的云平台上连接服务。

spring-boot-starter-data-elasticsearch

支持ElasticSearch搜索和分析引擎，包括spring-data-elasticsearch。

spring-boot-starter-data-gemfire

支持GemFire分布式数据存储，包括spring-data-gemfire。

spring-boot-starter-data-jpa

支持JPA（Java Persistence API. ，包括spring-data-jpa、spring-orm、Hibernate。

spring-boot-starter-data-mongodb

支持MongoDB数据，包括spring-data-mongodb。

spring-boot-starter-data-rest

通过spring-data-rest-webmvc，支持通过REST暴露Spring Data数据仓库。

spring-boot-starter-data-solr

支持Apache Solr搜索平台，包括spring-data-solr。

spring-boot-starter-freemarker

支持FreeMarker模板引擎。

spring-boot-starter-groovy-templates

支持Groovy模板引擎。

spring-boot-starter-hateoas

通过spring-hateoas支持基于HATEOAS的RESTful Web服务。

spring-boot-starter-hornetq

通过HornetQ支持JMS。

spring-boot-starter-integration

支持通用的spring-integration模块。

spring-boot-starter-jdbc

支持JDBC数据库。

spring-boot-starter-jersey

支持Jersey RESTful Web服务框架。

spring-boot-starter-jta-atomikos

通过Atomikos支持JTA分布式事务处理。

spring-boot-starter-jta-bitronix

通过Bitronix支持JTA分布式事务处理。

spring-boot-starter-mail

支持javax.mail模块。

spring-boot-starter-mobile

支持spring-mobile。

spring-boot-starter-mustache

支持Mustache模板引擎。

spring-boot-starter-redis

支持Redis键值存储数据库，包括spring-redis。

spring-boot-starter-security

支持spring-security。

spring-boot-starter-social-facebook

支持spring-social-facebook

spring-boot-starter-social-linkedin

支持pring-social-linkedin

spring-boot-starter-social-twitter

支持pring-social-twitter

spring-boot-starter-test

支持常规的测试依赖，包括JUnit、Hamcrest、Mockito以及spring-test模块。

spring-boot-starter-thymeleaf

支持Thymeleaf模板引擎，包括与Spring的集成。

spring-boot-starter-velocity

支持Velocity模板引擎。

spring-boot-starter-web

支持全栈式Web开发，包括Tomcat和spring-webmvc。

spring-boot-starter-websocket

支持WebSocket开发。

spring-boot-starter-ws

支持Spring Web Services。

Spring Boot应用启动器面向生产环境的还有2种，具体如下：

spring-boot-starter-actuator

增加了面向产品上线相关的功能，比如测量和监控。

spring-boot-starter-remote-shell

增加了远程ssh shell的支持。 最后，Spring Boot应用启动器还有一些替换技术的启动器，具体如下：

spring-boot-starter-jetty

引入了Jetty HTTP引擎（用于替换Tomcat. 。

spring-boot-starter-log4j

支持Log4J日志框架。

spring-boot-starter-logging

引入了Spring Boot默认的日志框架Logback。

spring-boot-starter-tomcat

引入了Spring Boot默认的HTTP引擎Tomcat。

spring-boot-starter-undertow

引入了Undertow HTTP引擎（用于替换Tomcat. 。

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<optional>true</optional> //表示开启热部署

</dependency>

</dependencies>

支持热部署，需要配置的依赖包，除此之外还需要插件spring-boot-maven-plugin

1. 配置依赖包和插件

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

<version>2.0.0.RELEASE</version>

<configuration>

<fork>true</fork>

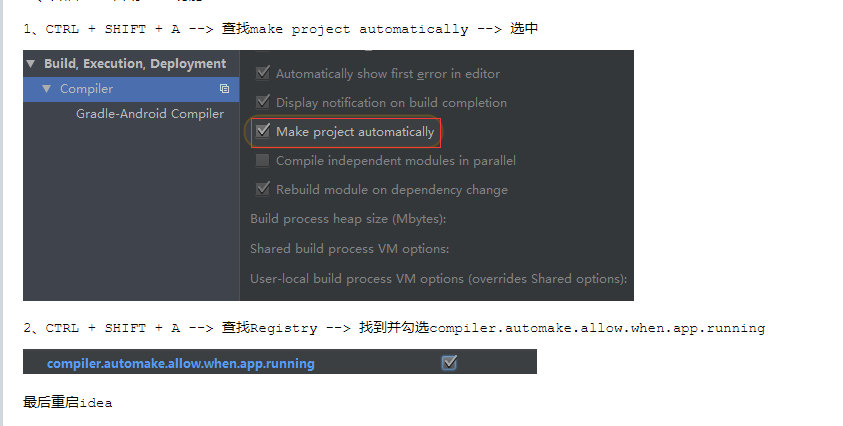
</configuration>

</plugin>

</plugins>

</build>

1. 开启idea功能



# 插件配置

1. Spring-boot-maven-plugin

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

作用：

1. Spring Boot的Maven插件（Spring Boot Maven plugin）能够以Maven的方式为应用提供Spring Boot的支持，即**为Spring Boot应用提供了执行Maven操作的可能**。
2. Spring Boot Maven plugin能够**将Spring Boot应用打包为可执行的jar或war文件，然后以通常的方式运行Spring Boot应用。**
3. Spring Boot Maven plugin的5个Goals

spring-boot:repackage，默认goal。在mvn package之后，再次打包可执行的jar/war，同时保留mvn package生成的jar/war为.origin

spring-boot:run，运行Spring Boot应用

spring-boot:start，在mvn integration-test阶段，进行Spring Boot应用生命周期的管理

spring-boot:stop，在mvn integration-test阶段，进行Spring Boot应用生命周期的管理

spring-boot:build-info，生成Actuator使用的构建信息文件build-info.properties

1. maven-source-plugin

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-source-plugin</artifactId>

<executions>

<execution>

<phase>package</phase>

<goals>

<goal>jar-no-fork</goal>

</goals>

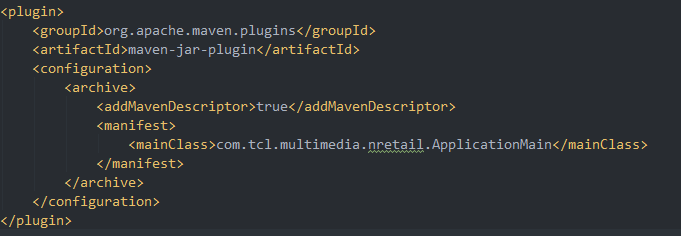
</execution>

</executions>

</plugin>

作用：

1. 需要对于Maven工程的源代码进行源文件的打包。可以利用source插件来完成。利用Maven的Source插件,对Maven工程的源码进行打jar包。
2. 将源文件打包
3. maven-jar-plugin



作用：

1. 用于指定本项目生成的jar包中的MANIFEST.MF文件中的配置，如Class-Path和Main-Class
2. <configuration>标签是作为DOM对象配置。
3. <manifest><mainClass> 用于指定运行的mainClass，
4. <manifest><addClasspath><classpathPrefix>标签用于将依赖的jar加入classpath中

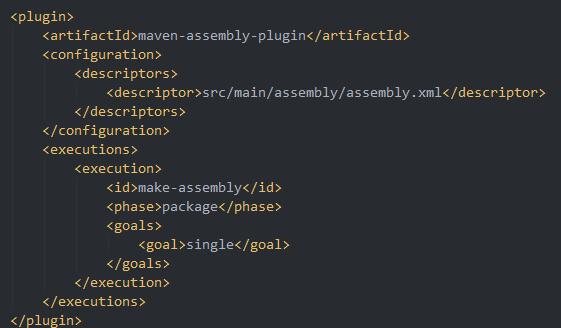


1. maven-compiler-plugin



作用：

1. maven是个项目管理工具，如果我们不告诉它我们的代码要使用什么样的jdk版本编译的话，它就会用maven-compiler-plugin默认的jdk版本来进行处理，这样就容易出现版本不匹配，以至于可能导致编译不通过的问题。
2. 使用maven-compiler-plugin插件可以指定项目源码的jdk版本，编译后的jdk版本，以及编码。
3. maven-assembly-plugin



作用：

1. 使用maven打包时，有三个相关的打包plugin



1. <descriptors>标签用来指定打包的配置文件，具体的打包方式在该配置文件中配置。