Creating Spring Boot Application

Terminologies –

Group Name : Package name of the application
Artifact : Name of the Application
Dependencies :
We would need to add spring boot starters and dependencies to your application inorder to do something.
Ex :
Spring boot starter web : Brings us lot of dependencies that are required to develop web applications.
DevTools : This dependency helps us to develop web applications in a easy and faster way.
spring-boot-starter-parent : Your application becomes the child of Spring boot starter parent (Parent pom providing dependency and plugin management for applications built with Maven) [It inherits lot of stuffs from spring boot starter parent project]
Build tool : Maven/Gradle project

Creating first Spring Boot Application. There are many ways to create Spring Boot Application.

1) Spring Initializr
2) Eclipse (Using STS Plugin) or STS (Spring Tool Suite)
3) Spring Boot CLI

1) Spring Initializr

The Spring Initializr is ultimately a web application that can generate a Spring Boot project structure for you.
Goto Spring initializr website https://start.spring.io/ to create a spring boot microservice Where
Group ID : com.example.Demo
Artifact Name : 01.Spring-boot-example
Select Java version which you are using
Click on Generate project and extract the zip file
Goto IDE (Integrated development environment) and Import it as Maven project or Gradle project.

Once the project is imported it downloads all the required libraries (Or dependencies) from Maven Repository.
There are many files and folders in this project like src/main/java (Where java codes are present), src/main/resources (Where property files are present), src/test/java (Where all test files are present) and pom.xml (Maven project) or build.gradle (Gradle project)

2) Eclipse (Using STS Plugin) or STS (Spring Tool Suite)

File -> New -> Spring Starter Project ->
Group Name : Package name of the application
Artifact : Name of the Application

3) Spring Boot CLI

It is a tool which you can download from the official site of Spring Framework. CLI executes groovy files So we can create and use groovy files to create Spring Boot Application

POM File
pom.xml
****

```
<?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
https://maven.apache.org/xsd/maven-4.0.0.xsd">
     <modelVersion>4.0.0</modelVersion>
     <parent>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-starter-parent</artifactId>
          <version>2.4.1</version>
          <relativePath/> <!-- lookup parent from repository
-->
     </parent>
     <groupId>com.example</groupId>
     <artifactId>01.Spring-boot-example</artifactId>
     <version>0.0.1-SNAPSHOT</version>
     <name>01.Spring-boot-example</name>
     <description>Demo project for Spring Boot</description>

     <properties>
          <java.version>11</java.version>
     </properties>

     <dependencies>
          <dependency>
               <groupId>org.springframework.boot</groupId>
               <artifactId>spring-boot-starter-web</
artifactId>
          </dependency>

          <dependency>
               <groupId>org.springframework.boot</groupId>
               <artifactId>spring-boot-devtools</artifactId>
               <scope>runtime</scope>
               <optional>true</optional>
          </dependency>
          <dependency>
               <groupId>org.projectlombok</groupId>
               <artifactId>lombok</artifactId>
               <optional>true</optional>
          </dependency>
          <dependency>
               <groupId>org.springframework.boot</groupId>
               <artifactId>spring-boot-starter-test</
artifactId>
               <scope>test</scope>
          </dependency>
     </dependencies>

     <build>
          <plugins>
               <plugin>
                    <groupId>org.springframework.boot</
groupId>
                    <artifactId>spring-boot-maven-plugin</
artifactId>
                    <configuration>
                         <excludes>
                              <exclude>

<groupId>org.projectlombok</groupId>
                                   <artifactId>lombok</
artifactId>
                              </exclude>
                         </excludes>
                    </configuration>
               </plugin>
          </plugins>
     </build>

</project>
```

****

Project description and packaging

So here groupId and artifactId are the ones which was given while creating spring boot project. Version (SNAPSHOT) means the project in dev mode.

Packaging is jar (Java Archive)

Typically in a web application we would use packaging called as war (Web Archive). Since it is a Spring boot application we would use packaging as jar because spring boot provides embedded server to achieve our purpose.

There are three sections

1) Parent - Specifying parent application so that we can inherit some features from it

2) Dependencies - All the dependencies located in folder called as Maven Dependencies

spring boot starter web - If you want to develop web application or rest application you need this dependency

Dev Tools - Brings features which makes developing applications easy

Spring boot starter test - Helps to write good unit test and integration test

Transitive Dependencies - We need some dependencies like starter web to develop web application, in same way starter web needs some other dependencies that is called as transitive dependency.
3) plugin - Spring boot maven plugin helps easily running spring boot applications and helps creating jar and war files out of spring boot web applications.
4) properties - Last one is Java Version

@SpringBootApplication annotation

@SpringBootApplication annotation initialises spring (Component Scan) and Spring boot (Auto configuration)
SpringApplication is a class that has a static method called run is used to run the spring boot application. It launches the application along with a server.

application.properties - Is a configuration file (Where lot of things can be configured)