Generating source with maven

Generate java from xml schema:

JAXB tutorial provides concepts and API to convert object into XML and XML into object. Our JAXB tutorial is designed for beginners and professionals.
JAXB stands for Java Architecture for XML Binding. It provides mechanism to marshal (write) java objects into XML and unmarshal (read) XML into object. Simply, you can say it is used to convert java object into xml and vice-versa.

Jaxb used to part of java standard distribution for java 8 and java 9 and now it is not.

maven-jaxb2-plugin is used to generate java code.

Xjc -> compiler used to generate sources.

Generate java from json schema:
Using jackson-databind for annotations…
Using commons-lang3 for equals and has function to be used by plugin.

Jackson itself doesn't directly use Apache Commons Lang (common-lang3) library. However, it's possible that a project using Jackson for JSON processing might also include Apache Commons Lang for additional utility functions, such as equals and hashcode methods.

```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
    <version>3.12.0</version>
  </dependency>
  <dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.9.1</version>
  </dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
```

```
      <groupId>org.jsonschema2pojo</groupId>
      <artifactId>jsonschema2pojo-maven-plugin</artifactId>
      <version>0.5.1</version>
      <configuration>
        <sourceDirectory>${basedir}/src/main/resources/schema</sourceDirectory>
        <targetPackage>pratik.maven.model</targetPackage>
        <useCommonsLang3>true</useCommonsLang3>
      </configuration>
      <executions>
        <execution>
          <goals>
            <goal>generate</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

Folder generated -> java-gen in target folder.

Lombok -> generate getters, setters, constructors, etc. from annotations.

Lombok.Data annotation ->
Gives getter, setter, toString, equals and hashcode.

MapStruct project -> create object using another object i.e. taking data from one pojo to another.

```
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.8.0</version>
            <configuration>
                <annotationProcessorPaths>
                    <path>
                        <groupId>org.mapstruct</groupId>
                        <artifactId>mapstruct-processor</artifactId>
                        <version>${mapstruct.version}</version>
                    </path>
                </annotationProcessorPaths>
            </configuration>
        </plugin>
    </plugins>
</build>
```

We are telling the compiler that we are hooking the map struct version.
Mapstruct-processor -> annotation processor used by compiler…