Building a Maven project

General view of the pom file

The structure of the project is described in the pom.xml file, which must be located in the root folder of the project. The content of the project file looks like this:

```
<project>
        <!—Description of the current project -->
        <groupId>...</groupId>
        <artifactId>...</artifactId>
        <packaging>...</packaging>
        <version>...</ version>

        <properties>
            <!-- Properties section -->
        </properties>

        <repositories>
            <!-- Repositories section -->
        </repositories>


        <dependencies>
            <!-- Dependencies section -->
        </dependencies>

        <build>
            <!-- Build section -->
        </build>
</project>
```

Not all sections may be present in the pom.xml description. So the properties and repositories sections are often not used. The description parameters of the current project are **required** . We will talk about the last section now.

build section

The build section is optional - Maven can build a project without it. But if you want to set up the assembly of a more or less complex project, then understanding how everything works there will come in handy.

Let's look at a simple example:

```
<build>
      <finalName>projectName</finalName>
      <sourceDirectory>${basedir}/src/java</sourceDirectory>
      <outputDirectory>${basedir}/targetDir</outputDirectory>
      <resources>
              <resource>
              <directory>${ basedir}/src/java/resources</directory>
              <includes>
                  <include>**/*.properties</include>
              </includes>
              </resource>
      </resources>
      <plugins>
              . . .
      </plugins>
  </build>
```

This section contains basic information about building: where Java files are located, resource files, what plugins are used, where to put the built project.

There are four main tags:

- <finalName>
- <sourceDirectory>
- <output directory>
- <resources>

Let's briefly analyze their purpose:

The <finalName> tag specifies the name of the resulting build file (jar, war, ear..) that is created in the **package** phase . If the parameter is not specified, then the default value, **artifactId-version ,** is used .

The <sourceDirectory> tag allows you to redefine the location of source files. By default, files are located in the **${basedir}/src/main/java** directory , but you can specify any other location.

The <outputDirectory> tag specifies the directory where the compiler will save the compilation results - ***.class** files. The default value **is target/classes** .

The <resources> tag and its nested <resource> tags define the location of the resource files. Resource files are simply copied into the **outputDirectory** directory when building . The default value of the resource directory is **src/main/resources** .

The build section can be configured particularly flexibly. We will look at it in more detail a little later.