

Painless organisation and maintenance of Apache Maven Builds

Sandra Parsick

🐙 @sparsick@mastodon.social

✉ mail@sandra-parsick.de

📍 Java Forum Stuttgart 09.07.2026

About me

- Sandra Parsick
- Freelance developer and consultant in Java ecosystem
- Project Lead Support & Care for Apache Maven
- Main topics:
 - Developing Java Applications in enterprise environment
 - Open Source
 - Software Craftmanship
 - Developer Experience / Plattform Engineering
- Trainings
- Workshops

✉ mail@sandra-parsick.de

🐙 [@sparsick@mastodon.social](https://mstdn.social/@sparsick)

🌐 <https://www.sandra-parsick.de>

🔗 <https://ready-for-review.dev>

About me

- Sandra Parsick
- Freelance developer and consultant in Java

✉ mail@sandra-parsick.de

🐙 [@sparsick@mastodon.social](https://mastodon.social/@sparsick)

🌐 <https://www.sandra-parsick.de>

👤 <https://ready-for-review.dev>

- Project Lead Support & Care for Apache Maven

- Development of Java Applications in enterprise

- Open Source

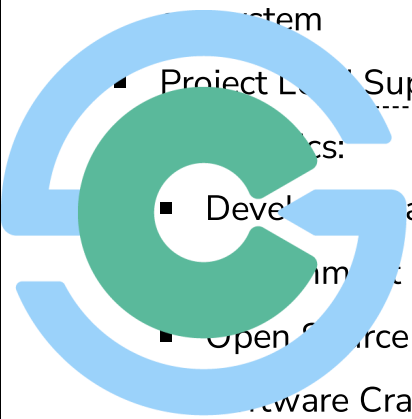
- Software Craftmanship

- Developer Experience / Plattform Engineering

- Trainings

- <https://open-elements.com/support-care>

- Workshops



Support & Care

by **Open**Elements

What Projects does Support & Care include



~ 50% marketshare in Java



> 2 billion downloads per year



> 75% marketshare in Java

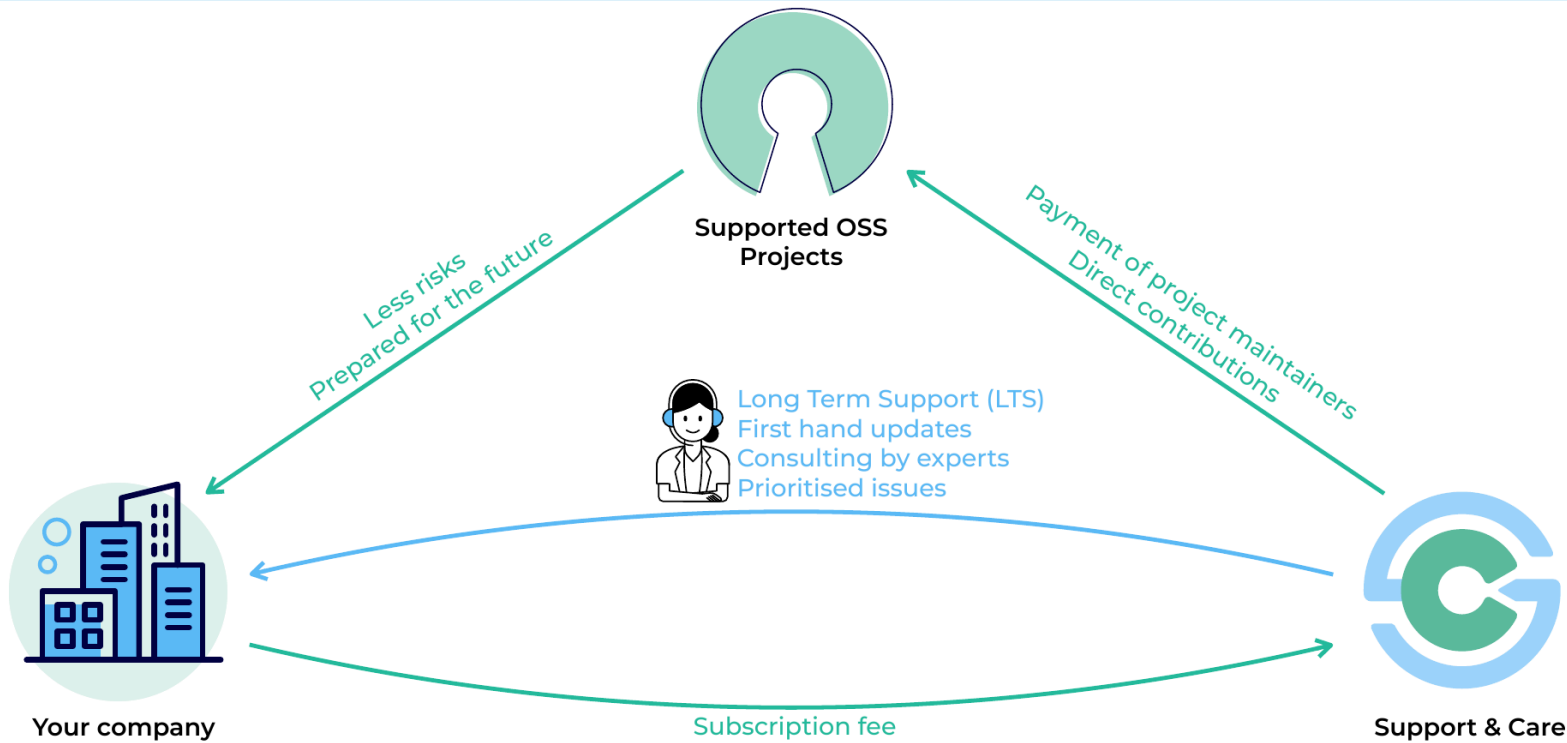


> 1 million downloads per month



> 500.000 downloads per day

... more to come



How Support & Care helps you and OSS

Your Benefits of Support & Care



Long Term Support (LTS) for most important versions that let you organize your updates better and do not let you use unsecured and unmaintained versions.



Fast fixes for bugs and information about security issues and updates on the newest versions



Individual workshops and consulting for the products done directly by the committers



Monthly Webinars on updates, the roadmap and important news and individual 1-on-1 Calls



Documentation, Support and Consulting in German and English by experts of the project



Custom builds and tools for the project done directly by the maintainers

Your Benefits of Support & Care

Agenda

- How to start a new Maven Project?

- Simplify Maintenance in Commons

- Dependency Maintenance

- Debugging Maven Builds



Long Term Support (LTS) for most important versions that let you organize your updates better and do not let you use unsecured and unmaintained versions.



Fast fixes for bugs and information about security issues and updates on the newest versions



Individual workshops and consulting for the products done directly by the committers



Monthly Webinars on updates, the roadmap and important news and individual 1-on-1 Calls



Documentation, Support and Consulting in German and English by experts of the project



Custom builds and tools for the project done directly by the maintainers

Problem: How to start a new Maven Project?

- How to start a new Maven Project?

▪ Simplify Maintenance in Commons

A commonly seen approach.

- Dependency Maintenance

Copy and paste of an already existing project.

- Debugging Maven Builds

Problem: How to start a new Maven Project?

- Outdated version are used
 - A lot of dependencies/configuration are inserted that maybe are not needed for my project
 - Copy and paste already existing maintenance problems
- Copy and paste of an already existing project.

What is the problem with this approach?

- Outdated version are used
- Plugins/dependency/configuration are inserted that maybe are not needed for my project
- Copy and paste already existing maintenance problems

Solution: We need a good Baseline
Based on a determininstic approach

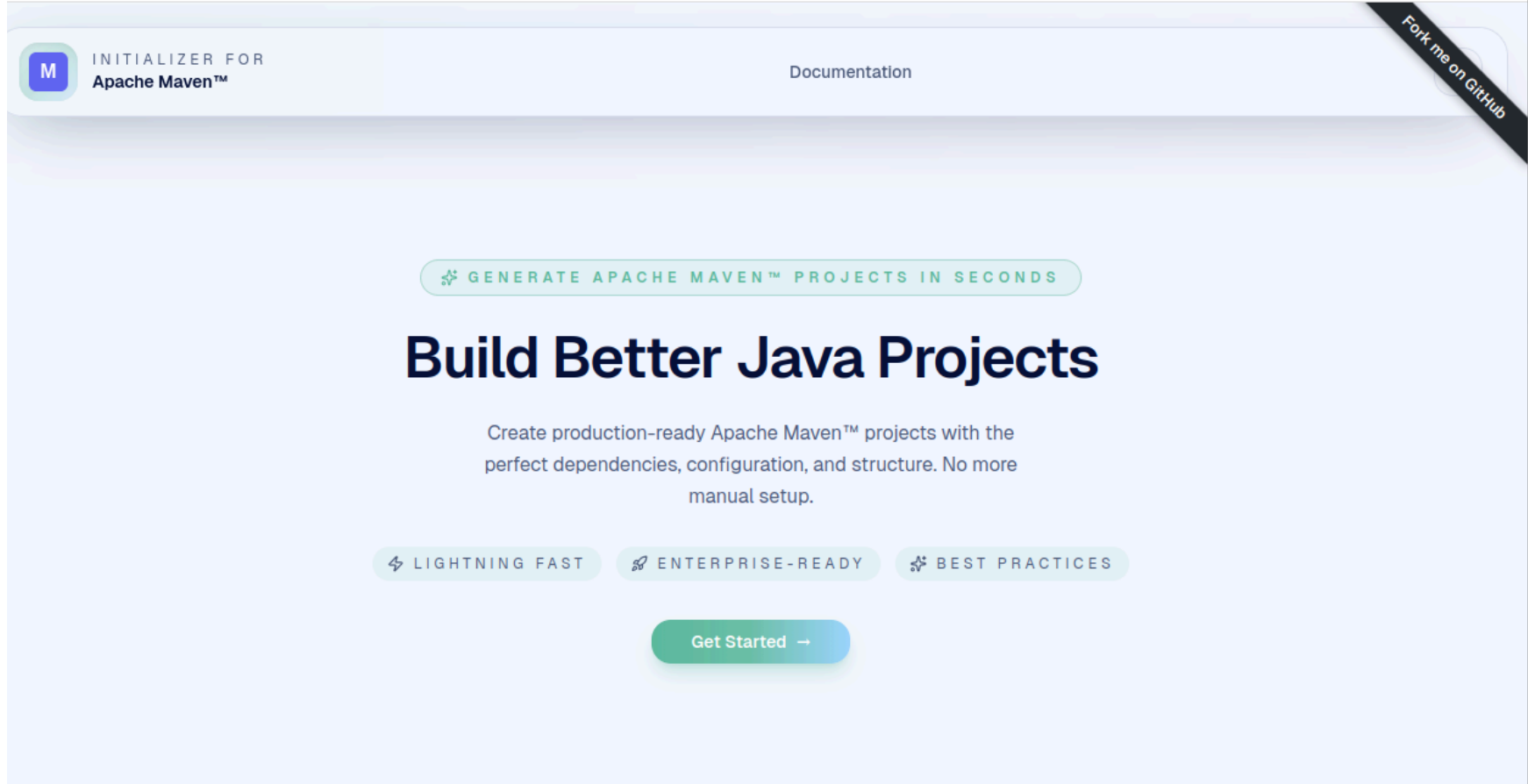
Possible Solutions

- Web-based approaches:
 - Initializer for Apache Maven
 - Spring Initializer
 - Microprofiler Starter (currently down for maintenance)
- CLI approaches:
 - Maven Archetypes
 - Maveniverse Toolbox

Solution: We need a good Baseline
Based on a deterministic approach

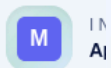
Potential Solution: Apache Maven

🌐 Web-based applications:
<https://maven.apache.org/>



Initial

https://



INITIALIZER FOR
Apache Maven™



INITIALIZER FOR
Apache Maven™

Documentation



PROJECT CONFIGURATION

CONFIGURE YOUR APACHE MAVEN™ PROJECT

Fill in the details below to generate your production-ready Apache Maven™ project structure.

Group ID *

com.example

Usually a domain name in reverse order

Artifact ID *

my-awesome-project

The name of your project

Version

1.0.0-SNAPSHOT

The version of your project

Project Name

My Awesome Project

A friendly name for your project

Java Version

Java 25 (Latest)

The Java version to use

Project Description

Describe your project's purpose and functionality...

A brief description of what your project does

ADVANCED OPTIONS

Configure additional options for your Apache Maven™ project.

☒ Include Apache Maven™ Wrapper

Includes mvnw and mvnw.cmd scripts for building without requiring Apache Maven™ to be installed.

Assertion Library

None (JUnit only)

Choose which assertion library to use in tests

Code Formatting Plugins

Select one or both plugins to ensure code style consistency.

☐ Spotless Maven Plugin

Keeps your code spotless by automatically formatting it. Requires configuration.

☐ Maven Checkstyle Plugin

Checks that your code adheres to a coding standard. Requires configuration.

Generate Project →

me on GitHub

Project Structure

```
.
├── mvnw
├── mvnw.cmd
├── pom.xml
├── README.md
└── src
    ├── main
    │   ├── java
    │   │   ├── com
    │   │   │   └── example
    │   │   │       └── Example.java
    └── test
        ├── java
        │   ├── com
        │   │   └── example
        │   │       └── ExampleTest.java
```

Generated POM

```
<?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/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>
  <artifactId>example</artifactId>
  <version>1.0.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <name/>
  <description/>
  <properties>
    <maven.compiler.release>25</maven.compiler.release>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.junit</groupId>
        <artifactId>junit-bom</artifactId>
        <version>6.0.3</version>
        <type>pom</type>
        <scope>import</scope>
```


Spring Initializer

 <https://start.spring.io/>



Project

- ☐ Gradle - Groovy
- ☐ Gradle - Kotlin
- ☒ Maven

Language

- ☒ Java ☐ Kotlin
- ☐ Groovy

Spring Boot

- ☐ 4.1.0 (SNAPSHOT) ☐ 4.1.0 (M4) ☐ 4.0.6 (SNAPSHOT)
- ☒ 4.0.5 ☐ 3.5.14 (SNAPSHOT) ☐ 3.5.13

Project Metadata

Group

Artifact

Package name

Packaging ☒ Jar ☐ War

Configuration ☒ Properties ☐ YAML

Java ☐ 26 ☒ 25 ☐ 21 ☐ 17

Dependencies

ADD DEPENDENCIES... CTRL + B

Spring Web WEB

Build web, including RESTful, applications using Spring MVC.
Uses Apache Tomcat as the default embedded container.



GENERATE CTRL + G

EXPLORE CTRL + SPACE

...

CLI Approaches - Maven Archetypes

```
→ mvn archetype:generate  
  -DarchetypeGroupId=org.apache.maven.archetypes  
  -DarchetypeArtifactId=maven-archetype-simple  
  -DarchetypeVersion=1.5
```



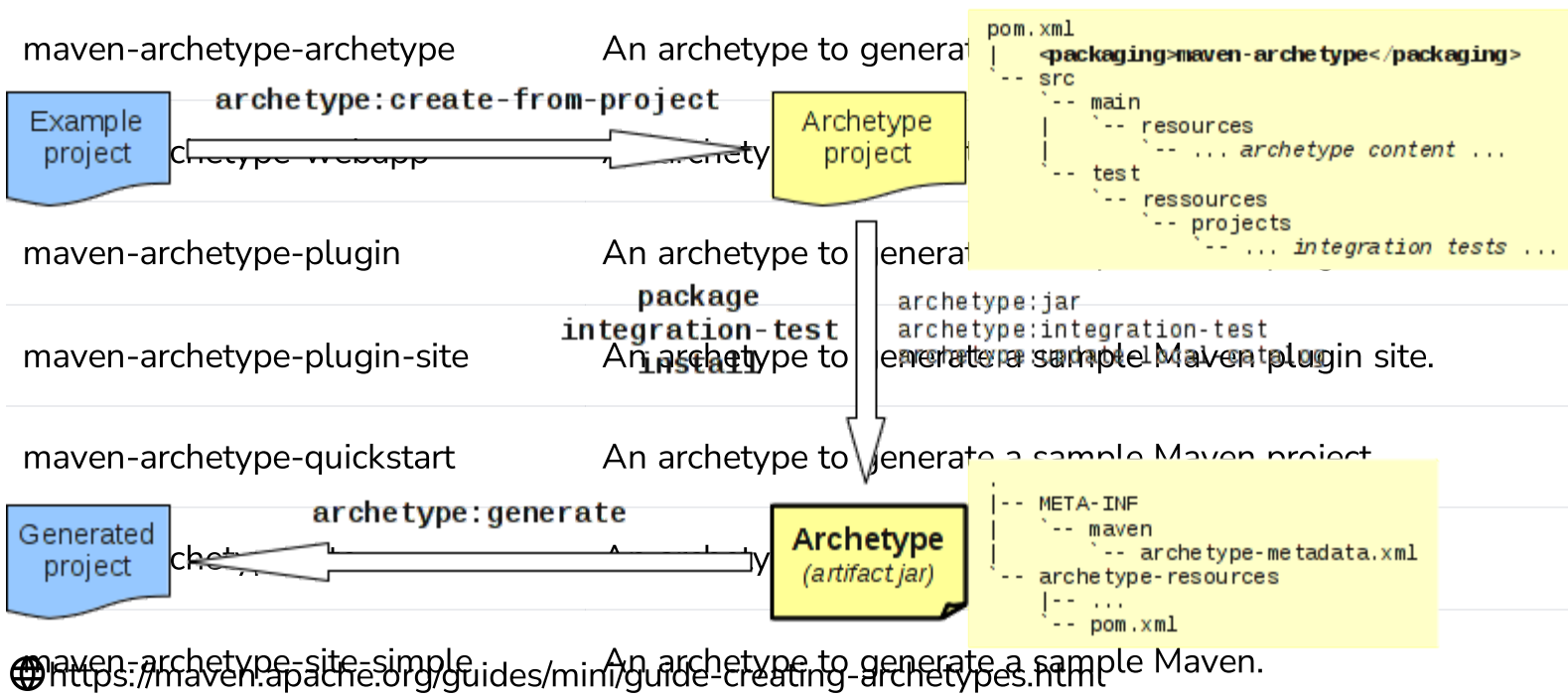
<https://maven.apache.org/guides/introduction/introduction-to-archetypes.html>

Maven Archetype Catalog (Maven Archetypes provided, a selection)

Archetype Artifact Ids	Description
maven-archetype-archetype	An archetype to generate a sample archetype.
maven-archetype-webapp	An archetype to generate a sample Maven Webapp project.
maven-archetype-plugin	An archetype to generate a sample Maven plugin. → mvn archetype:generate
maven-archetype-plugin-site	An archetype to generate a sample Maven plugin site. -DarchetypeGroupId=org.apache.maven.archetypes -DarchetypeArtifactId=maven-archetype-simple -DarchetypeVersion=1.5
maven-archetype-quickstart	An archetype to generate a sample Maven project. https://maven.apache.org/guides/introduction/introduction-to-archetypes.html
maven-archetype-site	An archetype to generate a sample Maven site.
maven-archetype-site-simple	An archetype to generate a sample Maven.
maven-archetype-simple	An archetype to generate a simple Maven project.

MarcheTypeCatalog (Apache Maven provided, a selection)

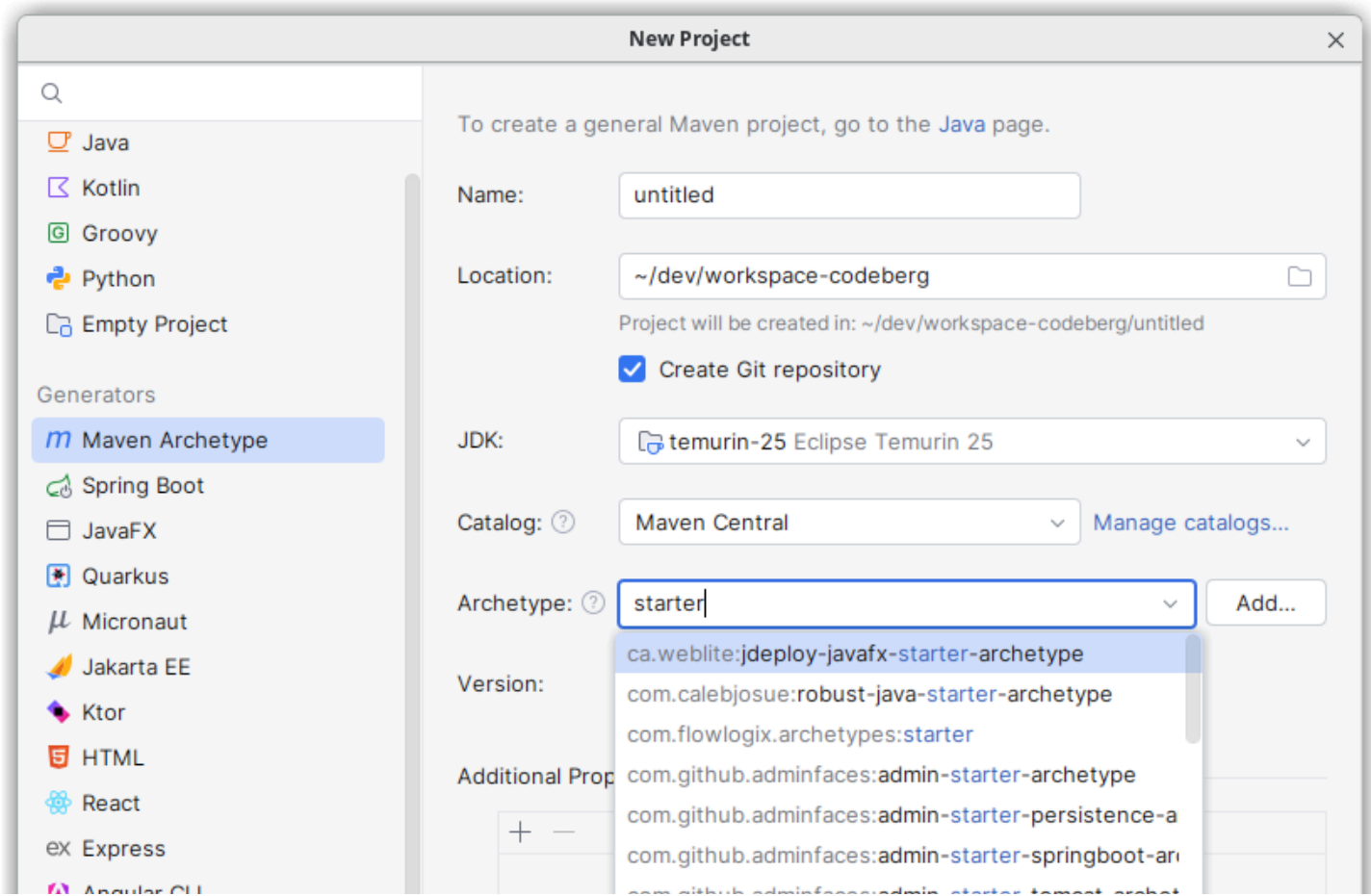
Archetype	Artifact Ids	Description
Useful, if you have a special project setup		



maven-archetype-simple

An archetype to generate a simple Maven project.

Maven Archetype is embedded in many dev tools





For more about IntelliJ Feature for Apache Maven,
check Marit's and Andres' talk Getting more out of Maven

New Kids on the block - Maveniverse Toolbox

 <https://maveniverse.eu/docs/toolbox/>

```
→ mvn toolbox:help
[INFO] --- toolbox:0.15.8:help (default-cli) @ standalone-pom ---
[INFO] eu.maveniverse.maven.plugins:toolbox 0.15.8
    Toolbox Maven Plugin and CLI in one.
```

This plugin has 84 goals: // a selection

```
toolbox:add-managed-dependency
    Adds managed dependency.
```

```
toolbox:add-managed-plugin
    Adds managed plugin.
```

```
toolbox:lock-plugin-versions
    Locks available versions of Maven Project used plugins.
```

```
toolbox:new-project
    Creates a new project.
```

```
toolbox:new-subproject
    Creates a new subproject.
```


ev/workspace-codeberg/sample-app via  v21.0.8 ...

▶ 0:00 / 0:56



Simplify Maintenance in Commons

- POM Formatting
- Pre-Checks to avoid build failure because of misconfiguration

▶ 0:00 / 0:56



Simplify Maintenance in Commons

- POM Formatting
- Pre-Checks to avoid build failure because of misconfiguration

POM Formatting

```
<?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"
```

```
<modelVersion>4.0.0</modelVersion>

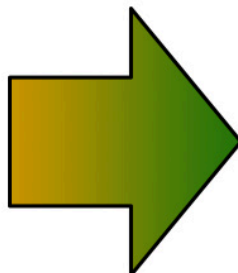
  <!-- Basics -->
  <groupId>com.github.ekryd.sortpom</groupId>
  <packaging>pom</packaging>
  <name>SortPom</name>
  <artifactId>sortpom-parent</artifactId>
  <version>2.6.1-SNAPSHOT</version>
  <description>A Maven plugin that sorts the pom.xml file.</description>
  <url>https://github.com/Ekryd/sortpom</url>
  <inceptionYear>2009</inceptionYear>

  <prerequisites>
    <maven>3.0.4</maven>
  </prerequisites>

  <properties>
    <compileSource>1.8</compileSource>
    <project.build.sourceEncoding>UTF-8
  </project.build.sourceEncoding>
  </properties>

  <modules>
    <module>sorter</module>
    <module>maven-plugin</module>
  </modules>

  <dependencies>
    <!-- Test dependencies -->
    <dependency>
      <artifactId>junit</artifactId>
      <groupId>junit</groupId>
      <version>4.12</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <version>1.3</version>
      <groupId>org.hamcrest</groupId>
      <artifactId>hamcrest-all</artifactId>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>com.github.ekryd.reflection-utils</groupId>
      <artifactId>reflection-utils</artifactId>
```



```
<?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"
```

```
<modelVersion>4.0.0</modelVersion>

  <!-- Basics -->
  <groupId>com.github.ekryd.sortpom</groupId>
  <artifactId>sortpom-parent</artifactId>
  <version>2.6.1-SNAPSHOT</version>
  <packaging>pom</packaging>
  <name>SortPom</name>
  <description>A Maven plugin that sorts the pom.xml file.</description>

  <prerequisites>
    <maven>3.0.4</maven>
  </prerequisites>

  <modules>
    <module>sorter</module>
    <module>maven-plugin</module>
  </modules>

  <properties>
    <compileSource>1.8</compileSource>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <dependencies>
    <!-- Test dependencies -->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.12</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.hamcrest</groupId>
      <artifactId>hamcrest-all</artifactId>
      <version>1.3</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>com.github.ekryd.reflection-utils</groupId>
      <artifactId>reflection-utils</artifactId>
      <version>1.1.0</version>
      <scope>test</scope>
    </dependency>
```

POM Formatting - Sortpom Maven Plugin

https://github.com/Ekryd/sortpom

```
<?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>

  <!-- Basics -->
  <groupId>com.github.ekryd.sortpom</groupId>
  <packaging>pom</packaging>
  <name>Sortpom</name>
  <artifactId>sortpom</artifactId>
  <version>1.0.0</version>
  <description>Sortpom</description>
  <url>https://github.com/Ekryd/sortpom</url>
  <inceptionYear>2013</inceptionYear>

  <prerequisites>
    <maven>3.0.4</maven>
  </prerequisites>

  <properties>
    <compileSource>1.8</compileSource>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <modules>
    <module>sorter</module>
    <module>maven-plugin</module>
  </modules>

  <dependencies>
    <!-- Test dependencies -->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.12</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.hamcrest</groupId>
      <artifactId>hamcrest-all</artifactId>
      <version>1.3</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>com.github.ekryd.reflection-utils</groupId>
      <artifactId>reflection-utils</artifactId>
      <version>1.0</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

Available Goals

→ mvn sortpom:sort
→ mvn sortpom:verify

Sort Build Configuration

```
<plugin>
  <groupId>com.github.ekryd.sortpom</groupId>
  <artifactId>sortpom-maven-plugin</artifactId>
  <version>3.3.0</version>
  <configuration>
    <createBackupFile>false</createBackupFile>
    <predefinedSortOrder>custom_1</predefinedSortOrder>
    <lineSeparator>\n</lineSeparator>
    <sortProperties>true</sortProperties>
    <sortDependencies>scope,groupId,artifactId</sortDependencies>
  </configuration>
  <executions>
    <execution>
      <goals>
        <goal>sort</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

POM Checkstyle Sortpom Maven Plugin of misconfiguration

🌐 <https://github.com/Ekryd/sortpom>

■ Is the correct Java version set?

Available Goals

■ Do you use the correct Maven version?

→ mvn sortpom:sort

Solution: Maven Enforcer Plugin

🌐 <https://maven.apache.org/enforcer>

Sort E

<plu

<g

<a

<v

<c

</

<e

</

</pl

enforcer plugin basic configuration

<build>

<plugins>

<plugin>

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

<artifactId>maven-enforcer-plugin</artifactId>

<version>3.6.2</version>

<executions>

<execution>

<id>enforcer</id>

<goals>

<goal>enforce</goal>

</goals>

<configuration>

<rules>

<!-- Add here your rules-->

</rules>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

Check, if the correct Java Version is used?

enforcer require Java version configuration

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-enforcer-plugin</artifactId>
  <version>3.6.2</version>
  <executions>
    <execution>
      <id>enforcer</id>
      <goals>
        <goal>enforce</goal>
      </goals>
      <configuration>
        <rules>
          <requireJavaVersion>
            <version>11,17,21</version>
          </requireJavaVersion>
        </rules>
      </configuration>
    </execution>
  </executions>
</plugin>
```

Check, if the correct Java Version is used?

if check fails

```
[INFO] --- enforcer:3.6.2:enforce (enforcer) @ testcontainers-gitea ---
[INFO] _____
[INFO] BUILD FAILURE
[INFO] _____
[INFO] Total time: 0.489 s
[INFO] Finished at: 2026-03-24T12:40:17Z
[INFO] _____
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-enforcer-plugin:3.6.2:
enforce (enforcer) on project testcontainers-gitea:
[ERROR] Rule 0: org.apache.maven.enforcer.rules.version.RequireJavaVersion failed with message:
[ERROR] Detected JDK /home/sparsick/.sdkman/candidates/java/8.0.472-tem/jre is version
1.8.0-472 which is not in the allowed range [11,17,21,).
[ERROR] → [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoExecutionException
```


Other rules, that could be helpful

Check, if the correct Java Version is used?

if check fails

requireJavaVendor :

Ensure, which Java Vendor can be used

```
<requireJavaVendor>
  <excludes>
    <exclude>Pivotal</exclude>
    <exclude>Oracle JDK</exclude>
  </excludes>
  <includes>
    <include>AdoptOpenJDK</include>
  </includes>
</requireJavaVendor>
```

1.0.0-472 which is not in the allowed range [11,17,21,).

[ERROR] → [Help 1]

[ERROR]

[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.

[ERROR] Re-run Maven using the -X switch to enable full debug logging.

[ERROR]

[ERROR] For more information about the errors and possible solutions, please read the following articles:

[ERROR] [Help 1] <http://cwiki.apache.org/confluence/display/MAVEN/MojoExecutionException>

requireOS :

or Ensure, which OS can be used

```
<requireOS>
— <name>Windows XP</name>
  <family>windows</family>
  <arch>x86</arch>
— <version>5.1.2600</version>
</requireOS>
```

tion.RequireJavaVersion failed with message:
tes/java/8.0.472-tem/jre is version

Check rules that could be enforced

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-enforcer-plugin</artifactId>
  <version>3.6.2</version>
  <executions>
    <execution>
      <id>enforcer</id>
      <goals>
        <goal>enforce</goal>
      </goals>
      <configuration>
        <rules>
          <requireJavaVersion>
            <version>11,17,21</version>
          </requireJavaVersion>
          <requireMavenVersion>
            <version>3.8</version>
          </requireMavenVersion>
        </rules>
      </configuration>
    </execution>
  </executions>
</plugin>
```

Check, if the correct Maven version is used?

```
[INFO] --- maven-enforcer-plugin:3.6.2:enforce (enforcer) @ testcontainers-gitea ---
[INFO] Rule 0: org.apache.maven.enforcer.rules.version.RequireJavaVersion passed
[INFO] _____
[INFO] BUILD FAILURE
[INFO] _____
[INFO] Total time: 0.903 s
[INFO] Finished at: 2026-03-24T13:32:40Z
[INFO] _____
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-enforcer-plugin:3.6.2:enforce (enforcer)
on project testcontainers-gitea:
[ERROR] Rule 1: org.apache.maven.enforcer.rules.version.RequireMavenVersion failed with message:
[ERROR] Detected Maven Version: 3.6.3 is not in the allowed range [3.8,).
[ERROR] → [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoExecutionException

    </configuration>
  </execution>
</executions>
</plugin>
```

Other possibility, to ensure the correct Maven version is used: *Maven Wrapper*

Maven Wrapper

🌐 <https://maven.apache.org/tools/wrapper/>

- Set Up:

```
mvn wrapper:wrapper
```

Other possibility, to ensure the correct M

```
→ tree -a
```

```
.
├── .gitignore
├── .mvn
│   └── wrapper
│       └── maven-wrapper.properties
├── mvnw
├── mvnw.cmd
├── pom.xml
├── README.md
└── src
```

Maven Wrapper Dependencies Maintenance

- 🌐 <https://maven.apache.org/tools/wrapper/>
- Dependency Management
- Set Up:
 - dependencies vs dependencyManagement

`mvn wrapper:wrapper`

- BOM vs properties

- Dependency Clean up
 - Find unused dependencies
 - Find using transitive dependencies
 - Avoid transitive dependencies
 - Banned dependencies
- Stay up-to-date with Your Dependencies
 - Versions Maven Plugin
 - renovate

```
→ tree -a
.
├── .gitignore
├── .mvn
│   └── wrapper
│       └── maven-wrapper.properties
├── mvnw
├── mvnw.cmd
├── pom.xml
├── README.md
└── src
```

Dependency Maintenance

`<dependencies>` vs `<dependencyManagement>`

- Dependency management

dependencies

```
<dependencies>
  <dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
    <version>3.20.0</version>
  </dependency>
</dependencies>
```

- Avoid transitive dependencies
- Banned dependencies
- Stay up-to-date with Your Dependencies
 - Versions Maven Plugin
 - renovate

dependencyManagement

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.apache.commons</groupId>
      <artifactId>commons-lang3</artifactId>
      <version>3.20.0</version>
    </dependency>
  </dependencies>
</dependencyManagement>
```

<!-- ... -->

```
<dependencies>
  <dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
  </dependency>
</dependencies>
```

Dependency Management

<dependencies> vs <dependencyManagement>

- define which dependency should be used during compile, test or execution/runtime (scoping).

- Use means this dependency is added to one or more specific classpaths

- Three kinds of classpath:

- compile classpath
 - `<groupId>org.apache.commons</groupId>`
 - `<artifactId>commons-lang3</artifactId>`
 - `<version>3.20.0</version>`
- test classpath
- execution classpath

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.apache.commons</groupId>
      <artifactId>commons-lang3</artifactId>
      <version>3.20.0</version>
    </dependency>
  </dependencies>
</dependencyManagement>
```

<!-- ... -->

```
<dependencies>
  <dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
  </dependency>
</dependencies>
```


Effects of Dependency Scoping

Scope	which classpath?	Special Features
■ define which dependency should be used during compile, test or execution/runtime (scoping).		
■ "Use" means this dependency is added to one or more specific classpaths		
■ (default) Three kinds of classpath		
■ compile classpath		
provided classpath	compile, test	like compile but you expect that a container will provide it at runtime
■ execution classpath		
runtime	test, execution	
test	test	
import	-	only type pom is supported
system	compile, test, execution	not provided via Maven Repository

Effects of <dependencyManagement>

Scope which classpath? Special Features

- without <dependencies> they have no effects.

■ compile *Single source of truth for versioning:*

(default) Useful especially in multi module projects

- enables using BOMs in Apache Maven

provided

compile, test

like compile but you expect that a container will provide it at runtime

runtime

test, execution

test

test

import

-

only type pom is supported

system

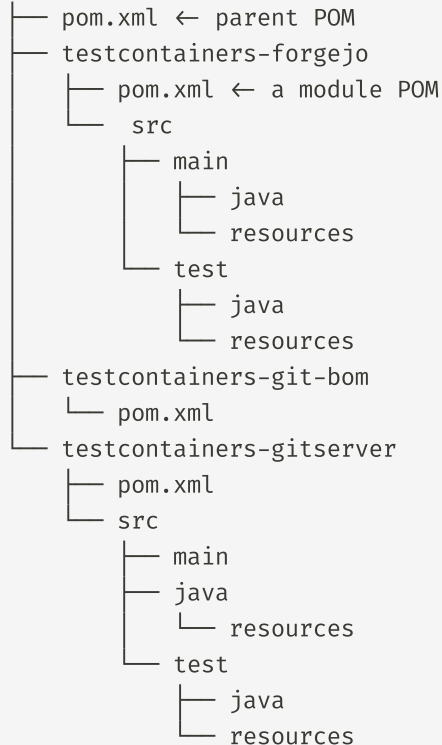
compile, test,
execution

not provided via Maven Repository

Effects of `<dependencyManagement>` and multi module projects

structure of multi module projects

- without `<dependencyManagement>`
- *Single source of truth*
 - Useful especially for large projects
 - enables using Maven



<dependencyManagement> and multi module projects

structure of multi module projects

snippet of parent POM

```
<groupId>dev.parsick.testcontainers.gitserver</groupId>
<artifactId>testcontainers-git-parent</artifactId>
<version>0.16.0-SNAPSHOT</version>
<packaging>pom</packaging>

<modules>
  <module>testcontainers-gitserver</module>
  <module>testcontainers-git-bom</module>
  <module>testcontainers-forgejo</module>
</modules>

<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.apache.commons</groupId>
      <artifactId>commons-lang3</artifactId>
      <version>3.20.0</version>
    </dependency>
  </dependencies>
</dependencyManagement>
```

snippet of a module POM

```
<parent>
  <groupId>dev.parsick.testcontainers.gitserver</groupId>
  <artifactId>testcontainers-git-parent</artifactId>
  <version>0.16.0-SNAPSHOT</version>
</parent>

<artifactId>testcontainers-forgejo</artifactId>

<dependencies>
  <dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
  </dependency>
</dependencies>
```

resources

<dependencyManagement> and BOMs
BOMs?

What is that? 🤔

BOM (Bill Of Material)

- origin idea coming from manufacturing area
- in software engineering: a list of all artifacts that are included to a library, framework, etc.
- for example, `junit` :
 - `junit-jupiter`
 - `junit-jupiter-api`
 - `junit-jupiter-engine`
 - `junit-jupiter-migrationsupport`
 - `junit-jupiter-params`
 - `junit-platform-commons`

1	2	3	4	5	6
Pos.	Menge	Einheit	Benennung	Sachnummer / Norm - Kurzbezeichnung	Bemerkung
1	1	Stck.	Gehäuse		G — AlSi10Mg
2	1	Stck.	Hohlwelle		34CrMo4
3	1	Stck.	Schneckenrad		G — CuSn12Ni
4	1	Stck.	Schneckenwelle		16MnCr5
5	1	Stck.	Zahnrad		16MnCr5
6	2	Stck.	Lagerdeckel groß		S235JR
7	1	Stck.	Lagerdeckel klein		S235JR
8	1	Stck.	Distanzring		S235JR
9	2	Stck.	Rillenkugellager	DIN 625 — 6009	
10	2	Stck.	Kegelrollenlager	DIN 720 — 30203	
11	1	Stck.	Passfeder groß	DIN 6885 — B 12 x 8 x 22	
12	1	Stck.	Passfeder klein	DIN 6885 — B 5 x 5 x 10	
13	2	Stck.	Verschlusschraube	DIN 908 — M14 x 15 — St	
14	2	Stck.	Dichtring	DIN 7603 — A 14 x 18 Vf	
15	15	Stck.	Zylinderschraube mit Innensechskant	ISO 4762 — M6 x 20 — 8.8	
16	1	Stck.	Zylinderschraube mit Innensechskant	ISO 4762 — M6 x 16 — 8.8	
17	1	Stck.	Scheibe	DIN 9021 — B 6,4	
18	2	Stck.	Radial-Wellendichtring	DIN 3760 — AS 4,5 x 60 x 8	
19	2	Stck.	O-Ring	DIN 3771-85x3,55-N-NBR 70	
20	1	Stck.	O-Ring	DIN 3771-40x3,55-N-NBR 70	
21	4	Stck.	Stiftschraube	Kaufteil gemäß Zeichnung	S235JR

		Datum		Name
		Bearb.	20.03.2006	
		Gepr.		
		Norm		
Zust.	Anforderung	Datum	Name	

Schneckengetriebe

Blatt
1
von **10 Bl.**

BOM (Bill of Materials) in Apache Maven?

POM definition of BOM artifact

```
<project>
  <modelVersion>4.0.0</modelVersion>

  <groupId>dev.parsick.testcontainers.gitserver</groupId>
  <artifactId>testcontainers-git-bom</artifactId>
  <version>0.16.0</version>
  <packaging>pom</packaging>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>dev.parsick.testcontainers.gitserver</groupId>
        <artifactId>testcontainers-gitserver</artifactId>
        <version>${project.version}</version>
      </dependency>
      <dependency>
        <groupId>dev.parsick.testcontainers.gitserver</groupId>
        <artifactId>testcontainers-forgejo</artifactId>
        <version>${project.version}</version>
      </dependency>
    </dependencies>
  </dependencyManagement>
</project>
```

How to use a BOM artifact in Apache Maven?

Using BOM definition in a project

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>dev.parsick.testcontainers.gitserver</groupId>
      <artifactId>testcontainers-git-bom</artifactId>
      <version>0.16.0</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
<dependencies>
  <dependency>
    <groupId>dev.parsick.testcontainers.gitserver</groupId>
    <artifactId>testcontainers-gitserver</artifactId>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>dev.parsick.testcontainers.gitserver</groupId>
    <artifactId>testcontainers-forgejo</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```


How to use a BOM artifact in Apache Maven?

Using BOM definition in a project

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>dev.parsick.testcontainers.gitserver</groupId>
      <artifactId>testcontainers-git-bom</artifactId>
      <version>0.16.0</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```

But what if I use a library that have many artifacts but no BOM artifact

Solution: Using <properties>

```
<dependencies>
  <dependency>
    <groupId>dev.parsick.testcontainers.gitserver</groupId>
    <artifactId>testcontainers-gitserver</artifactId>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>dev.parsick.testcontainers.gitserver</groupId>
    <artifactId>testcontainers-forgejo</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```

Using of `<properties>`

Option: Using directly

```
<properties>
  <jgit.version>7.6.0.202603022253-r</jgit.version>
</properties>

<dependencies>
  <dependency>
    <groupId>org.eclipse.jgit</groupId>
    <artifactId>org.eclipse.jgit</artifactId>
    <version>${jgit.version}</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.eclipse.jgit</groupId>
    <artifactId>org.eclipse.jgit.ssh.jsch</artifactId>
    <version>${jgit.version}</version>
    <scope>test</scope>
  </dependency>
</dependencies>
```

Using of `<properties>`

Option: Using via dependencyManagement

```
<properties>
  <jgit.version>7.6.0.202603022253-r</jgit.version>
</properties>

<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.eclipse.jgit</groupId>
      <artifactId>org.eclipse.jgit</artifactId>
      <version>${jgit.version}</version>
    </dependency>
    <dependency>
      <groupId>org.eclipse.jgit</groupId>
      <artifactId>org.eclipse.jgit.ssh.jsch</artifactId>
      <version>${jgit.version}</version>
    </dependency>
  </dependencies>
</dependencyManagement>
```

```
<dependencies>
  <dependency>
    <groupId>org.eclipse.jgit</groupId>
    <artifactId>org.eclipse.jgit</artifactId>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.eclipse.jgit</groupId>
    <artifactId>org.eclipse.jgit.ssh.jsch</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```

Dependency Clean up

Option: Using via dependencyManagement

Problem: How to find unused or using transitive dependencies

Maven Dependency Plugin

🌐 <https://maven.apache.org/plugins/maven-dependency-plugin/>

```
<properties>
```

```
<jgit.version>7.0.0.202003022255</jgit.version>
```

```
</properties>
```

```
<dependencyManagement>
```

```
<dependencies>
```

```
<dependency>
```

```
<groupId>org.eclipse.jgit</groupId>
```

```
<artifactId>org.eclipse.jgit</artifactId>
```

```
<version>${jgit.version}</version>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>org.eclipse.jgit</groupId>
```

```
<artifactId>org.eclipse.jgit.ssh</artifactId>
```

```
<version>${jgit.version}</version>
```

```
</dependency>
```

```
</dependencies>
```

```
</dependencyManagement>
```

```
<dependencies>
```

```
<dependency>
```

```
<groupId>org.eclipse.jgit</groupId>
```

```
<artifactId>org.eclipse.jgit</artifactId>
```

→ mvn dependency:analyze-only

[INFO] --- dependency:3.10.0:analyze-only (default-cli) @ testcontainers-gitea ---

[WARNING] Used undeclared dependencies found:

[WARNING] org.junit.jupiter:junit-jupiter-api:jar:6.0.3:test

[WARNING] org.apache.httpcomponents.core5:httpcore5:jar:5.4:compile

[WARNING] org.junit.jupiter:junit-jupiter-params:jar:6.0.3:test

[WARNING] com.jcraft:jsch:jar:0.1.55:test

[WARNING] commons-io:commons-io:jar:2.20.0:compile

[WARNING] org.jetbrains:annotations:jar:17.0.0:compile

[WARNING] com.github.docker-java:docker-java-api:jar:3.7.1:compile

[WARNING] Unused declared dependencies found:

[WARNING] ch.qos.logback:logback-classic:jar:1.5.32:provided

[WARNING] org.slf4j:slf4j-api:jar:2.0.17:compile

[WARNING] com.google.code.findbugs:jsr305:jar:3.0.2:compile

[WARNING] org.junit.jupiter:junit-jupiter:jar:6.0.3:test

[WARNING] Non-test scoped test only dependencies found:

[WARNING] org.jetbrains:annotations:jar:17.0.0:compile

Dependency Cleanup

build configuration

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-dependency-plugin</artifactId>
  <version>3.10.0</version>
  <executions>
    <execution>
      <goals>
        <goal>analyze-only</goal>
      </goals>
      <configuration>
        <failOnWarning>true</failOnWarning>
      </configuration>
    </execution>
  </executions>
</plugin>
```

[WARNING]

[WARNING]

[WARNING]

[WARNING]

[WARNING]

Non-te

org.jetbrains:annotations:jar:17.0.0:compile

→ mvn verify

...

[INFO] --- dependency:3.10.0:analyze-only (default) @ testcontainers-gitea

[ERROR] Used undeclared dependencies found:

[ERROR] org.junit.jupiter:junit-jupiter-api:jar:6.0.3:test

[ERROR] org.apache.httpcomponents.core5:httpcore5:jar:5.4:compile

[ERROR] org.junit.jupiter:junit-jupiter-params:jar:6.0.3:test

[ERROR] com.jcraft:jsch:jar:0.1.55:test

[ERROR] commons-io:commons-io:jar:2.20.0:compile

[ERROR] org.jetbrains:annotations:jar:17.0.0:compile

[ERROR] com.github.docker-java:docker-java-api:jar:3.7.1:compile

[ERROR] Unused declared dependencies found:

[ERROR] ch.qos.logback:logback-classic:jar:1.5.32:provided

[ERROR] org.slf4j:slf4j-api:jar:2.0.17:compile

[ERROR] com.google.code.findbugs:jsr305:jar:3.0.2:compile

[ERROR] org.junit.jupiter:junit-jupiter:jar:6.0.3:test

[ERROR] Non-test scoped test only dependencies found:

[ERROR] org.jetbrains:annotations:jar:17.0.0:compile

[INFO]

[INFO] BUILD FAILURE

[INFO]

Check in part of my build

Problems:

build configuration

→ mvn verify

→ mvn verify

...

[INFO] --- dependency:3.10.0:analyze-only (default) @ testcontainers-gitea ---

[ERROR] Used undeclared dependencies found:

[ERROR] org.junit.jupiter:junit-jupiter-api:jar:6.0.3:test ← 2)

[ERROR] org.apache.httpcomponents.core5:httpcore5:jar:5.4:compile

[ERROR] org.junit.jupiter:junit-jupiter-params:jar:6.0.3:test

[ERROR] com.jcraft:jsch:jar:0.1.55:test

[ERROR] commons-io:commons-io:jar:2.20.0:compile ← 1)

[ERROR] org.jetbrains:annotations:jar:17.0.0:compile

[ERROR] com.github.docker-java:docker-java-api:jar:3.7.1:compile

[ERROR] Unused declared dependencies found:

[ERROR] ch.qos.logback:logback-classic:jar:1.5.32:provided

[ERROR] org.slf4j:slf4j-api:jar:2.0.17:compile

[ERROR] com.google.code.findbugs:jsr305:jar:3.0.2:compile

[ERROR] org.junit.jupiter:junit-jupiter:jar:6.0.3:test ← 2)

[ERROR] Non-test scoped test only dependencies found:

[ERROR] org.jetbrains:annotations:jar:17.0.0:compile

[INFO] -----

[INFO] BUILD FAILURE

[INFO] -----

rs-gitea

le

e

Problem 1: commons-io

Easy: Add the dependency explicitly to your build configuration

```
<dependencies>
  <dependency>
    <groupId>commons-io</groupId>
    <artifactId>commons-io</artifactId>
  </dependency>
</dependencies>
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>commons-io</groupId>
      <artifactId>commons-io</artifactId>
      <version>2.20.0</version>
    </dependency>
  </dependencies>
</dependencyManagement>
```

```
[ERROR]    org.junit.jupiter:junit-jupiter:jar:6.0.3:test ← 2)
```

```
[ERROR] Non-test scoped test only dependencies found:
```

```
[ERROR]    org.jetbrains:annotations:jar:17.0.0:compile
```

```
[INFO] -----
```

```
[INFO] BUILD FAILURE
```

```
[INFO] -----
```

Problem 2: junit-jupiter

Starting point: dependency explicitly to your build configuration

```
<dependencies>
```

Dependency declaration

```
<dependencies>
```

```
<dependency>
```

```
<groupId>org.junit.jupiter</groupId>
```

```
<artifactId>junit-jupiter</artifactId>
```

```
<scope>test</scope>
```

```
</dependency>
```

```
</dependencies>
```

```
<dependencyManagement>
```

```
<dependencies>
```

```
<dependency>
```

```
<groupId>org.junit</groupId>
```

```
<artifactId>junit-bom</artifactId>
```

```
<version>6.0.3</version>
```

```
<type>pom</type>
```

```
<scope>import</scope>
```

```
</dependency>
```

```
</dependencies>
```

```
</dependencyManagement>
```

→ mvn verify

...

[INFO] --- dependency:3.10.0:analyze-only (default) @ testcontainers-gitea ---

[ERROR] Used undeclared dependencies found:

[ERROR] org.junit.jupiter:junit-jupiter-api:jar:6.0.3:test ← 2)

[ERROR] Unused declared dependencies found:

[ERROR] org.junit.jupiter:junit-jupiter:jar:6.0.3:test ← 2)

Problem 2: junit-jupiter

Starting point:

```
→ mvn dependency:tree
[INFO] Scanning for projects...
[INFO]
[INFO] ---< dev.parsick.testcontainers.gitserver:testcontainers-gitea >---
[INFO] Building Gitea Server for Testcontainers 0.16.0-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- dependency:3.10.0:tree (default-cli) @ testcontainers-gitea ---
[INFO] dev.parsick.testcontainers.gitserver:testcontainers-gitea:jar:0.16.0-SNAPSHOT
[INFO] +- org.junit.jupiter:junit-jupiter:jar:6.0.3:test <---
[INFO] | +- org.junit.jupiter:junit-jupiter-api:jar:6.0.3:test <---
[INFO] | | +- org.opentest4j:opentest4j:jar:1.3.0:test
[INFO] | | +- org.junit.platform:junit-platform-commons:jar:6.0.3:test
[INFO] | | +- org.apiguardian:apiguardian-api:jar:1.1.2:test
[INFO] | | \- org.jspecify:jspecify:jar:1.0.0:test
[INFO] | +- org.junit.jupiter:junit-jupiter-params:jar:6.0.3:test
[INFO] | \- org.junit.jupiter:junit-jupiter-engine:jar:6.0.3:test
[INFO] | \- org.junit.platform:junit-platform-engine:jar:6.0.3:test
</dependencies>
</dependencyManagement>
```

Problem 2: junit-jupiter

~~Explain~~ **Analyze**: junit-jupiter is an Aggregator dependency

```
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter</artifactId>
<version>6.0.3</version>
<name>JUnit Jupiter (Aggregator)</name>
<dependencies>
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-api</artifactId>
  </dependency>
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-params</artifactId>
  </dependency>
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-engine</artifactId>
    <scope>runtime</scope>
  </dependency>
</dependencies>
```

Problem 2: junit-jupiter

Explanation: junit-jupiter is an Aggregator dependency

1. Don't use aggregator dependencies 🙈 or
2. Give Maven Dependency Plugin more information:

```
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter</artifactId>
<version>5.0.3</version>

<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-dependency-plugin</artifactId>
  <executions>
    <execution>
      <configuration>
        <failOnWarning>true</failOnWarning>
        <ignoredUsedUndeclaredDependencies>
          <ignoredUsedUndeclaredDependency>org.junit.jupiter</ignoredUsedUndeclaredDependency>
        </ignoredUsedUndeclaredDependencies>
        <ignoredUnusedDeclaredDependencies>
          <ignoredUnusedDeclaredDependency>org.junit.jupiter</ignoredUnusedDeclaredDependency>
        </ignoredUnusedDeclaredDependencies>
      </configuration>
    </execution>
  </executions>
</plugin>
```

Maven Dependency Plugin configuration

Maven Dependency Plugin configuration

```
<configuration>
  <failOnWarning>true</failOnWarning>
  <!-- ignore jsr305 for "used but undeclared", "declared but unused", and "non-test scoped" -->
  <ignoredDependencies>
    <ignoredDependency>com.google.code.findbugs:jsr305</ignoredDependency>
  </ignoredDependencies>
  <!-- ignore annotations for "used but undeclared" warnings -->
  <ignoredUsedUndeclaredDependencies>
    <ignoredUsedUndeclaredDependency>com.google.code.findbugs:annotations</ignoredUsedUndeclaredDependency>
  </ignoredUsedUndeclaredDependencies>
  <!-- ignore annotations for "unused but declared" warnings -->
  <ignoredUnusedDeclaredDependencies>
    <ignoredUnusedDeclaredDependency>com.google.code.findbugs:annotations</ignoredUnusedDeclaredDependency>
  </ignoredUnusedDeclaredDependencies>
  <!-- ignore annotations for "non-test scoped" warnings -->
  <ignoredNonTestScopedDependencies>
    <ignoredNonTestScopedDependency>com.google.code.findbugs:annotations</ignoredNonTestScopedDependency>
  </ignoredNonTestScopedDependencies>
</configuration>
```

Maven DependencyToolbox can help analyzing dependency related problems

```
→ mvn toolbox:help
[INFO] eu.maveniverse.maven.plugins.toolbox 0.15.8
Toolbox Maven Plugin and CLI in one.

This plugin has 84 goals: // a selection

toolbox:classpath
  Prints the classpath of current project.

toolbox:classpath-list
  Prints the classpath of current project as list of artifacts.

toolbox:plugin-tree
  Collects project given plugin and output its dependency tree.

toolbox:project-dependency-tree
  Displays project interdependencies of Maven Projects as tree (forest).

toolbox:tree
  Displays dependency tree of Maven Project.
```

Dependency-Check can help analyzing dependency related problems

Banned (transitive) dependencies

```
[INFO] eu.maveniverse.maven.plugins:toolbox 0.15.8
```

```
Toolbox Maven Plugin and CLI in one
```

Use Case: Ensure that a specific dependency is not used anymore

```
This plugin has 84 goals: // a selection
```

Solution: Maven Enforcer Rules

```
toolbox:classpath
```

- **bannedDependencies**
Prints the classpath of current project.

- **banTransitiveDependencies**

```
toolbox:classpath-tree
```

Prints the classpath of current project as list of artifacts.

```
toolbox:plugin-tree
```

Collects project given plugin and output its dependency tree.

```
toolbox:project-dependency-tree
```

Displays project interdependencies of Maven Projects as tree (forest).

```
toolbox:tree
```

Displays dependency tree of Maven Project.

bannedDependencies vs banTransitiveDependencies

Banned (transitive) dependencies

- `bannedDependencies` ensure, that a dependency is not declared directly

Use Case: Ensure that a specific dependency is not used anymore

- `banTransitiveDependencies` ensure that a dependency is not fetched to your project transitively

Solution: Maven Enforcer Rules

- `bannedDependencies`
- `banTransitiveDependencies`

Maven Enforcer Rules and Dependencies

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-enforcer-plugin</artifactId>
  <version>3.6.2</version>
  <executions>
    <execution>
      <id>enforcer</id>
      <goals>
        <goal>enforce</goal>
      </goals>
      <configuration>
        <rules>
          <bannedDependencies>
            <excludes>
              <exclude>junit:junit</exclude>
            </excludes>
            ⚠ but includes of specific version is possible
          </bannedDependencies>
        </rules>
      </configuration>
    </execution>
  </executions>
</plugin>
```


Maven Enforcer Rule: bannedDependencies

```
→ mvn initialize
[INFO] Scanning for projects...
[INFO]
[INFO] —< dev.parsick.testcontainers.gitserver:testcontainers-forgejo >—
[INFO] Building Forgejo Server for Testcontainers 0.16.0-SNAPSHOT
[INFO] from pom.xml
[INFO] _____[ jar ]_____
[INFO]
[INFO] --- enforcer:3.6.2:enforce (enforcer) @ testcontainers-forgejo ---
[INFO] Rule 0: org.apache.maven.enforcer.rules.version.RequireJavaVersion passed
[INFO] Rule 1: org.apache.maven.enforcer.rules.version.RequireMavenVersion passed
[INFO] _____
[INFO] BUILD FAILURE
[INFO] _____
[INFO] Total time: 0.525 s
[INFO] Finished at: 2026-04-13T19:30:54+02:00
[INFO] _____
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-enforcer-plugin:3.6.2:enforce (enforcer) on project testconta
[ERROR] Rule 2: org.apache.maven.enforcer.rules.dependency.BannedDependencies failed with message:
[ERROR] dev.parsick.testcontainers.gitserver:testcontainers-forgejo:jar:0.16.0-SNAPSHOT
[ERROR] junit:junit:jar:4.13.2 <— banned via the exclude/include list
  </executions>
</plugin>
```

Maven Enforcer Rule: banTransitiveDependencies

Tree Structure Dependencies

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-enforcer-plugin</artifactId>
  <version>3.6.2</version>
  <executions>
    <execution>
      <id>enforcer</id>
      <goals>
        <goal>enforce</goal>
      </goals>
      <configuration>
        <rules>
          <banTransitiveDependencies>
            <excludes>
              <exclude>org.apache.commons:commons-compress</exclude>
            </excludes>
            <!-- but includes of specific version is possible -->
          </banTransitiveDependencies>
        </rules>
      </configuration>
    </execution>
  </executions>
</plugin>
```

Maven Enforcer Rule: banTransitiveDependencies

```
→ mvn initialize
```

```
[INFO] Scanning for projects...
```

```
[INFO]
```

```
[INFO] —< dev.parsick.testcontainers.gitserver:testcontainers-forgejo >—
```

```
[INFO] Building Forgejo Server for Testcontainers 0.16.0-SNAPSHOT
```

```
[INFO] from pom.xml
```

```
[INFO] ————— [ jar ] —————
```

```
[INFO]
```

```
[INFO] --- enforcer:3.6.2:enforce (enforcer) @ testcontainers-forgejo ---
```

```
[INFO] Rule 0: org.apache.maven.enforcer.rules.version.RequireJavaVersion passed
```

```
[INFO] Rule 1: org.apache.maven.enforcer.rules.version.RequireMavenVersion passed
```

```
[INFO] —————
```

```
[INFO] BUILD FAILURE
```

```
[INFO] —————
```

```
[INFO] Total time: 0.429 s
```

```
[INFO] Finished at: 2026-04-14T15:54:43+02:00
```

```
[INFO] —————
```

```
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-enforcer-plugin:3.6.2:enforce (enforcer) on project testcontainers-forgejo: Rule 2: org.apache.maven.enforcer.rules.dependency.BanTransitiveDependencies failed with message:
```

```
[ERROR] Rule 2: org.apache.maven.enforcer.rules.dependency.BanTransitiveDependencies failed with message:
```

```
[ERROR] dev.parsick.testcontainers.gitserver:testcontainers-forgejo:jar:0.16.0-SNAPSHOT
```

```
[ERROR] org.testcontainers:testcontainers:jar:2.0.4 has transitive dependencies:
```

```
[ERROR] org.apache.commons:commons-compress:jar:1.28.0 has transitive dependencies:
```

```
< plugin>
```

How to Enforce Related Transitive Dependencies in my project?

→ `mvn initialize`

exclusion configuration

```
<dependency>
  <groupId>org.testcontainers</groupId>
  <artifactId>testcontainers</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.apache.commons</groupId>
      <artifactId>commons-compress</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

[INFO]

[INFO] Total time: 0.429 s

[INFO] Finished at: 2026-04-14T15:54:43+02:00

[INFO]

[ERROR] Failed to execute goal org.apache.maven.plugins:maven-enforcer-plugin:3.6.2:enforce (**enforcer**) on project testconta

[ERROR] Rule 2: org.apache.maven.enforcer.rules.dependency.BanTransitiveDependencies failed with message:

[ERROR] dev.parsick.testcontainers.gitserver:testcontainers-forgejo:jar:0.16.0-SNAPSHOT

[ERROR] org.testcontainers:testcontainers:jar:2.0.4 has transitive dependencies:

[ERROR] org.apache.commons:commons-compress:jar:1.28.0 has transitive dependencies:

Stay up-to-date with Your Dependencies

■ Maven Way: Versions Maven Plugin

■ Third Party App: Renovate

```
<groupId>org.testcontainers</groupId>
<artifactId>testcontainers</artifactId>
<exclusions>
  <exclusion>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-compress</artifactId>
  </exclusion>
</exclusions>
</dependency>
```

Versions Maven Plugin

🌐 <https://www.mojohaus.org/versions/versions-maven-plugin/index.html>

```
→ mvn versions:display-dependency-updates
[INFO] Scanning for projects ...
[INFO]
[INFO] —< dev.parsick.testcontainers.gitserver:testcontainers-forgejo >—
[INFO] Building Forgejo Server for Testcontainers 0.16.0-SNAPSHOT
[INFO]    from pom.xml
[INFO] _____[ jar ]_____
[INFO]
[INFO] --- versions:2.21.0:display-dependency-updates (default-cli) @ testcontainers-forgejo ---
[INFO] No dependencies in Dependencies have newer versions.
[INFO]
[INFO] The following dependencies in Dependency Management have newer versions:
[INFO]   com.fasterxml.jackson.core:jackson-annotations ..... 2.21 → 3.0-rc5
[INFO]   org.assertj:assertj-core ..... 3.27.7 → 4.0.0-M1
[INFO]   org.junit.jupiter:junit-jupiter ..... 6.0.3 → 6.1.0-M1
[INFO]   org.junit.jupiter:junit-jupiter-api ..... 6.0.3 → 6.1.0-M1
[INFO]   org.junit.jupiter:junit-jupiter-engine ..... 6.0.3 → 6.1.0-M1
[INFO]   org.junit.jupiter:junit-jupiter-migrationsupport ... 6.0.3 → 6.1.0-M1
[INFO]   org.junit.jupiter:junit-jupiter-params ..... 6.0.3 → 6.1.0-M1
[INFO]   org.junit.platform:junit-platform-commons ..... 6.0.3 → 6.1.0-M1
[INFO]   org.junit.platform:junit-platform-console ..... 6.0.3 → 6.1.0-M1
[INFO]   org.junit.platform:junit-platform-engine ..... 6.0.3 → 6.1.0-M1
[INFO]   org.junit.platform:junit-platform-launcher ..... 6.0.3 → 6.1.0-M1
[INFO]   org.junit.platform:junit-platform-reporting ..... 6.0.3 → 6.1.0-M1
[INFO]   org.junit.platform:junit-platform-suite ..... 6.0.3 → 6.1.0-M1
```

Version Management

Using Maven Dependency Management, we need a little bit of configuration

🌐 <https://www.mojohaus.org/versions/versions-maven-plugin/index.html>

version-maven-plugin configuration

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>versions-maven-plugin</artifactId>
  <version>2.21.0</version>
  <configuration>
    <processDependencyManagementTransitive>>false</processDependencyManagementTransitive>
  </configuration>
</plugin>
```

[INFO]

[INFO] The following dependencies in Dependency Management have newer versions:

[INFO] com.fasterxml.jackson.core:jackson-annotations 2.21 → 3.0-rc5

[INFO] org.assertj:assertj-core 3.27.7 → 4.0.0-M1

[INFO] org.junit.jupiter:junit-jupiter 6.0.3 → 6.1.0-M1

[INFO] org.junit.jupiter:junit-jupiter-api 6.0.3 → 6.1.0-M1

[INFO] org.junit.jupiter:junit-jupiter-engine 6.0.3 → 6.1.0-M1

[INFO] org.junit.jupiter:junit-jupiter-migrationsupport ... 6.0.3 → 6.1.0-M1

[INFO] org.junit.jupiter:junit-jupiter-params 6.0.3 → 6.1.0-M1

[INFO] org.junit.platform:junit-platform-commons 6.0.3 → 6.1.0-M1

[INFO] org.junit.platform:junit-platform-console 6.0.3 → 6.1.0-M1

[INFO] org.junit.platform:junit-platform-engine 6.0.3 → 6.1.0-M1

[INFO] org.junit.platform:junit-platform-launcher 6.0.3 → 6.1.0-M1

[INFO] org.junit.platform:junit-platform-reporting 6.0.3 → 6.1.0-M1

[INFO] org.junit.platform:junit-platform-suite 6.0.3 → 6.1.0-M1

If we are using dependency management, we need a little bit of configuration

```
→ mvn versions:display-dependency-updates
```

```
[INFO] —< dev.parsick.testcontainers.gitserver:testcontainers-git-parent >—
[INFO] Building Git Server for Testcontainers 0.16.0-SNAPSHOT [1/6]
[INFO] from pom.xml
[INFO] —————[ pom ]—————
[INFO]
[INFO] --- versions:2.21.0:display-dependency-updates (default-cli) @ testcontainers-git-parent ---
[INFO] The following dependencies in Dependency Management have newer versions:
[INFO] org.assertj:assertj-core ..... 3.27.7 → 4.0.0-M1
[INFO] org.junit:junit-bom ..... 6.0.3 → 6.1.0-M1
[INFO] org.openrewrite.recipe:rewrite-recipe-bom ..... 3.26.1 → 3.28.0
[INFO] org.slf4j:slf4j-api ..... 2.0.17 → 2.1.0-alpha1
[INFO]
...
[INFO] —< dev.parsick.testcontainers.gitserver:testcontainers-forgejo >—
[INFO] Building Forgejo Server for Testcontainers 0.16.0-SNAPSHOT [5/6]
[INFO] from testcontainers-forgejo/pom.xml
[INFO] —————[ jar ]—————
[INFO]
[INFO] --- versions:2.21.0:display-dependency-updates (default-cli) @ testcontainers-forgejo ---
[INFO] The following dependencies in Dependencies have newer versions:
[INFO] com.fasterxml.jackson.core:jackson-annotations ..... 2.21 → 3.0-rc5
[INFO] org.assertj:assertj-core ..... 3.27.7 → 4.0.0-M1
[INFO] org.junit.jupiter:junit-jupiter ..... 6.0.3 → 6.1.0-M1
[INFO] org.slf4j:slf4j-api ..... 2.0.17 → 2.1.0-alpha1
[INFO]
```


versions:display-plugin-updates ~~at the end of the configuration~~

→ mvn versions:display-plugin-updates

```
[INFO] --- versions:2.21.0:display-plugin-updates (default-cli) @ testcontainers-git-parent ---
```

```
[INFO]
```

```
[INFO] The following plugin updates are available:
```

```
[INFO]   com.github.spotbugs:spotbugs-maven-plugin ..... 4.9.8.2 → 4.9.8.3
```

```
[INFO]
```

```
[INFO] All plugins have a version specified.
```

```
[INFO]
```

```
[INFO] Project requires minimum Maven version for build of: 3.8
```

```
[INFO] Plugins require minimum Maven version of: 3.6.3
```

```
[INFO]
```

```
[INFO] No plugins require a newer version of Maven than specified by the pom.
```

```
[INFO]
```

```
[INFO] Require Maven 4.0.0-beta-3 to use the following plugin updates:
```

```
[INFO]   maven-compiler-plugin ..... 3.15.0 → 4.0.0-beta-1
```

```
[INFO]   maven-resources-plugin ..... 3.5.0 → 4.0.0-beta-1
```

```
[INFO]   maven-source-plugin ..... 3.4.0 → 4.0.0-beta-1
```

```
[INFO]
```

```
[INFO] Require Maven 4.0.0-rc-2 to use the following plugin updates:
```

```
[INFO]   maven-compiler-plugin ..... 3.15.0 → 4.0.0-beta-2
```

```
[INFO]
```

```
[INFO] Require Maven 4.0.0-rc-4 to use the following plugin updates:
```

```
[INFO]   maven-compiler-plugin ..... 3.15.0 → 4.0.0-beta-4
```

```
[INFO]
```

```
[INFO]   org.slf4j:slf4j-api ..... 2.0.17 → 2.1.0-alpha1
```

```
[INFO]
```

Also Maveniverse Toolbox can help with the search for **plugins** updates

```
→ mvn toolbox:help
```

```
[INFO] eu.maveniverse.maven.plugins:toolbox 0.15.8
```

```
Toolbox Maven Plugin and CLI in one.
```

```
This plugin has 84 goals: // a selection
```

```
toolbox:dependency-versions
```

```
Lists available versions of Maven Project dependencies.
```

```
toolbox:plugin-versions
```

```
Lists available versions of Maven Project plugins.
```

```
toolbox:versions
```

```
Lists available updates to everything of Maven Project (including core extensions).
```

```
[INFO] maven-resources-plugin ..... 3.5.0 → 4.0.0-beta-1
```

```
[INFO] maven-source-plugin ..... 3.4.0 → 4.0.0-beta-1
```

```
[INFO]
```

```
[INFO] Require Maven 4.0.0-rc-2 to use the following plugin updates:
```

```
[INFO] maven-compiler-plugin ..... 3.15.0 → 4.0.0-beta-2
```

```
[INFO]
```

```
[INFO] Require Maven 4.0.0-rc-4 to use the following plugin updates:
```

```
[INFO] maven-compiler-plugin ..... 3.15.0 → 4.0.0-beta-4
```

```
[INFO]
```

Dependency updates for Maven / plugin updates



🌐 <https://docs.renovatebot.com/>

`toolbox:plugins-versions`

Lists available versions of Maven Project plugins.

`toolbox:versions`

Lists available updates to everything of Maven Project (including core extensions).

Supported Platforms:

- Azure DevOps
- AWS CodeCommit
- Bitbucket Cloud
- Bitbucket Server
- Forgejo
- Gerrit (experimental)
- Gitea
- GitHub (.com and Enterprise Server)
- GitLab (.com and CE/EE)

Label issues and pull requests for new contributors

[Dismiss](#)

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with **good first issue**

[Go to Labels](#)

Filters ▾

Labels 7

Milestones 0

[New pull request](#)☐ 13 Open ✓ 149 Closed

Author ▾

Label ▾

Projects ▾

Milestones ▾

Reviews ▾

Assignee ▾

Sort ▾

☐  **Update dependency org.apache.maven.plugins:maven-project-info-reports-plugin to v3.4.1** ✓

#310 opened 25 days ago by renovate (bot) 1 task

☐  **Update dependency org.apache.maven.plugins:maven-jxr-plugin to v3.3.0** ✓

#309 opened 26 days ago by renovate (bot) 1 task

☐  **Update dependency org.apache.maven.plugins:maven-javadoc-plugin to v3.4.1** ✓

#308 opened 26 days ago by renovate (bot) 1 task

1

☐  **Update dependency org.eclipse.platform:org.eclipse.osgi to v3.18.200** ✓

#292 opened on May 31 by renovate (bot) 1 task

☐  **Update dependency org.codehaus.plexus:plexus-component-metadata to v2.1.1** ✓

#291 opened on May 31 by renovate (bot) 1 task

1

☐  **Update dependency org.codehaus.plexus:plexus-component-annotations to v2.1.1** ✓

#290 opened on May 31 by renovate (bot) 1 task

1

☐  **Update dependency org.apache.maven.plugins:maven-plugin-plugin to v3.7.0** ✓

#284 opened on Dec 16, 2021 by renovate (bot) 1 task

1

☐  **Update dependency org.apache.maven.plugin-tools:maven-plugin-annotations to v3.7.0** ✓

#265 opened on Apr 26, 2021 by renovate (bot) 1 task

1

Update dependency org.apache.maven.plugins:maven-project-info-reports-plugin to v3.4.1 #310

[Edit](#)[Code](#)[Gitpod](#)

renovate wants to merge 1 commit into [master](#) from [renovate/org.apache.maven.plugins-maven-project-info-reports-plugin-3.x](#)

[Conversation](#) 0[Commits](#) 1[Checks](#) 8[Files changed](#) 1[+1 -1](#)

renovate bot commented 25 days ago

Contributor



This PR contains the following updates:

Package	Change	Age	Adoption	Passing	Confidence
org.apache.maven.plugins:maven-project-info-reports-plugin	3.1.1 -> 3.4.1	124d	32%	96%	high

Configuration

Schedule: Branch creation - At any time (no schedule defined), Automerge - At any time (no schedule defined).

Automerge: Disabled by config. Please merge this manually once you are satisfied.

Rebasing: Whenever PR becomes conflicted, or you tick the rebase/retry checkbox.

Ignore: Close this PR and you won't be reminded about this update again.

☐ If you want to rebase/retry this PR, check this box

This PR has been generated by [Mend Renovate](#). View repository job log [here](#).



Update dependency org.apache.maven.plugins:maven-project-info-reports...

Verified

✓ 13bcd79

Reviewers

Suggestions

[Request](#)

Still in progress? Convert to draft

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

Notifications

Customize



Unsubscribe

You're receiving notifications because you're watching this repository.

Update dependency org.apache.maven.plugins:maven-project-info-reports-plugin to v3.4.1 #310

Debugging Maven Builds

Open renovate wants to merge 1 commit into master from renovate/org.apache.maven.plugins-maven-project-info-reports-plugin-3.x

■ Verbose Log output

■ Buildtime Analyse

■ Remote Debug

Contributor

MENDRenovate

This PR contains the following updates:

Package	Change	Age	Adoption	Passing	Confidence
org.apache.maven.plugins:maven-project-info-reports-plugin	3.1.1 -> 3.4.1	124d	32%	96%	high

Configuration

Schedule: Branch creation - At any time (no schedule defined), Automerge - At any time (no schedule defined).

Automerge: Disabled by config. Please merge this manually once you are satisfied.

Rebasing: Whenever PR becomes conflicted, or you tick the rebase/retry checkbox.

Ignore: Close this PR and you won't be reminded about this update again.

☐ If you want to rebase/retry this PR, check this box

This PR has been generated by [Mend Renovate](#). View repository job log [here](#).

Update dependency org.apache.maven.plugins:maven-project-info-reports...

Reviewers

Suggestions

Still in progress? Convert to draft

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

Notifications

Customize

Edit

<> Code

Gitpod

Unsubscribe

You're receiving notifications because you're watching this repository.

Debugging Maven Builds

Using CLI Flag `-X`
Verbose Log `-put`

```
→ mvn -X initialize
Apache Maven 3.9.14 (996c630dbc656c76214ce58821dcc58be960875b)
Maven home: /home/sparsick/.sdkman/candidates/maven/current
Java version: 21.0.8, vendor: Eclipse Adoptium, runtime: /home/sparsick/.sdkman/candidates/java/21.0.8-tem
Default locale: de_DE, platform encoding: UTF-8
OS name: "linux", version: "6.17.0-112020-tuxedo", arch: "amd64", family: "unix"
[DEBUG] Created new class realm maven.api
[DEBUG] Importing foreign packages into class realm maven.api
[DEBUG]   Imported: javax.annotation.* < plexus.core
[DEBUG]   Imported: javax.annotation.security.* < plexus.core
...
[DEBUG] Looking up lifecycle mappings for packaging pom from ClassRealm[plexus.core, parent: null]
[DEBUG] == REACTOR BUILD PLAN ==
[DEBUG] Project: dev.parsick.testcontainers.gitserver:testcontainers-forgejo:jar:0.16.0-SNAPSHOT
[DEBUG] Tasks:    [initialize]
[DEBUG] Style:     Regular
[DEBUG] =====
[INFO] —< dev.parsick.testcontainers.gitserver:testcontainers-forgejo >—
[INFO] Building Forgejo Server for Testcontainers 0.16.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] ————— [ jar ] —————
```

Where is my build time gone?

maven-buildtime-profiler Extension

 <https://github.com/khmarbaise/maven-buildtime-profiler>

Define a `.mvn/extensions.xml` file in your project:

extension configuration

```
<extensions xmlns="http://maven.apache.org/EXTENSIONS/1.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/EXTENSIONS/1.1.0
    http://maven.apache.org/xsd/core-extensions-1.1.0.xsd">
  <extension>
    <groupId>com.soebes.maven.extensions</groupId>
    <artifactId>maven-buildtime-profiler</artifactId>
    <version>0.5.0</version>
  </extension>
</extensions>
```


Where is my build time gone?

BuildTimeReporter - profiler Extension

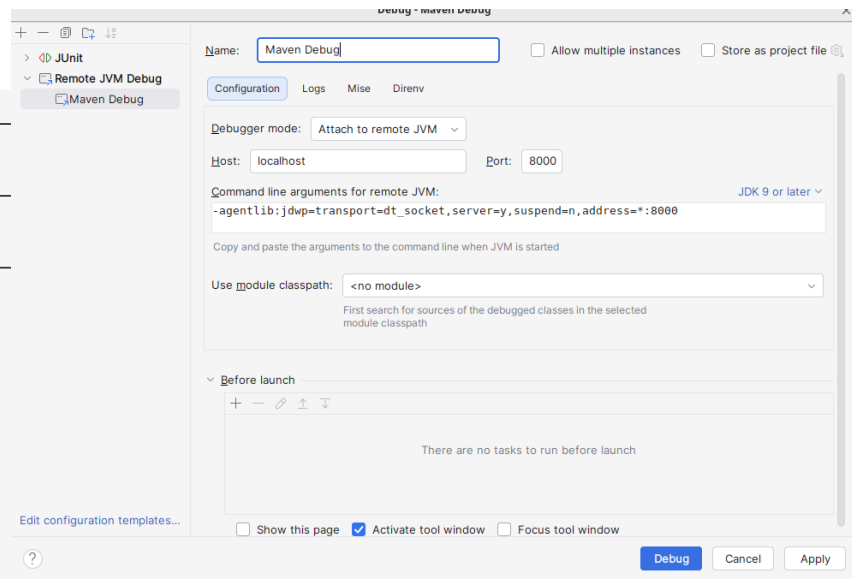
```
[INFO] -----  
[INFO] --                Maven Build Time Profiler Summary                --  
[INFO] -----  
[INFO] Project discovery time:           65 ms  
[INFO] -----  
[INFO] Project Build Time (reactor order):  
[INFO]  
[INFO] Git Server for Testcontainers:  
[INFO]           155 ms : validate  
[INFO]           63 ms : initialize  
[INFO] Git Servers for Testcontainers:  
[INFO]           2 ms : validate  
[INFO]           2 ms : initialize  
[INFO]           36 ms : process-resources  
[INFO]           51 ms : compile  
[INFO]           3 ms : process-test-resources  
[INFO]           4 ms : test-compile  
[INFO]          82343 ms : test  
[INFO] Rewrite for Git Server for Testcontainers:  
[INFO]           2 ms : validate  
[INFO]           0 ms : initialize  
[INFO]           2 ms : process-resources
```

Where Remote Debugging?

Buildtime Report

Command `mvnDebug` :

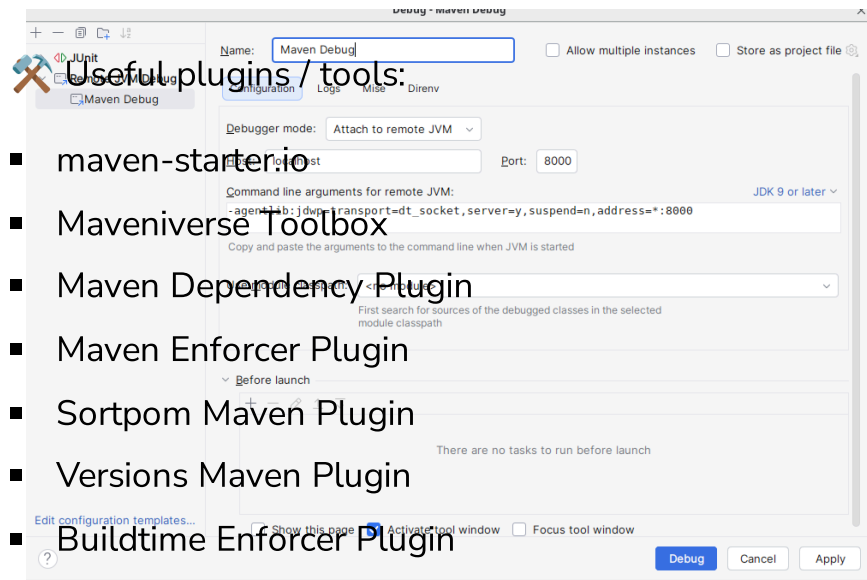
```
[INFO] -----  
→ mvnDebug test  
Preparing to execute Maven in debug mode  
Listening for transport dt_socket at address: 8000  
[INFO] -----  
[INFO] Project Build Time (reactor order):  
[INFO]  
[INFO] Git Server for Testcontainers:  
[INFO]      155 ms : validate  
[INFO]      63 ms : initialize  
[INFO] Git Servers for Testcontainers:  
[INFO]      2 ms : validate  
[INFO]      2 ms : initialize  
[INFO]      36 ms : process-resources  
[INFO]      51 ms : compile  
[INFO]      3 ms : process-test-resources  
[INFO]      4 ms : test-compile  
[INFO]      82343 ms : test  
[INFO] Rewrite for Git Server for Testcontainers:  
[INFO]      2 ms : validate  
[INFO]      0 ms : initialize  
[INFO]      2 ms : process-resources
```



Slammy Remote Debugging

💡 Command mynDebug :

```
→ mvnDebug test  
Preparing to execute Maven in debug mode  
Listening for transport dt_socket at address: 8000
```



My Maven Training @ socreatory



Apache Maven Masterclass

Build stable, traceable, and maintainable build processes with Maven.

Check out all details



<https://www.socreatory.com/trainings/maven-master>



Get **20%** off the regular ticket price with the code:

MAVEN_ROCKS

My Maven Training @ socreatory



Apache Maven Masterclass

Questions?

 <https://codeberg.org/sparsick/maven-maintenance-talk>

 mail@sandra-parsick.de

 [@sparsick@mastodon.social](https://mstdn.social/@sparsick)

Check out all details



<https://www.socreatory.com/trainings/maven-master>

Get **20%** off the regular ticket price with the code:

MAVEN_ROCKS

