All you need to know about Maven 4

Maven™

About me

Matthias Bünger

- 🧑‍💻 Software engineer
- 🧑‍🔧 OSS: JUnit Pioneer, Maven
- 𝕏 @bukamabish
- @bukama@mastodon.social
- 🦋 @bukama.bsky.social

What Maven version are you using?

Agenda

- Motivation
- Major changes Maven 4
- Improved user experience
- Glimpse into the future

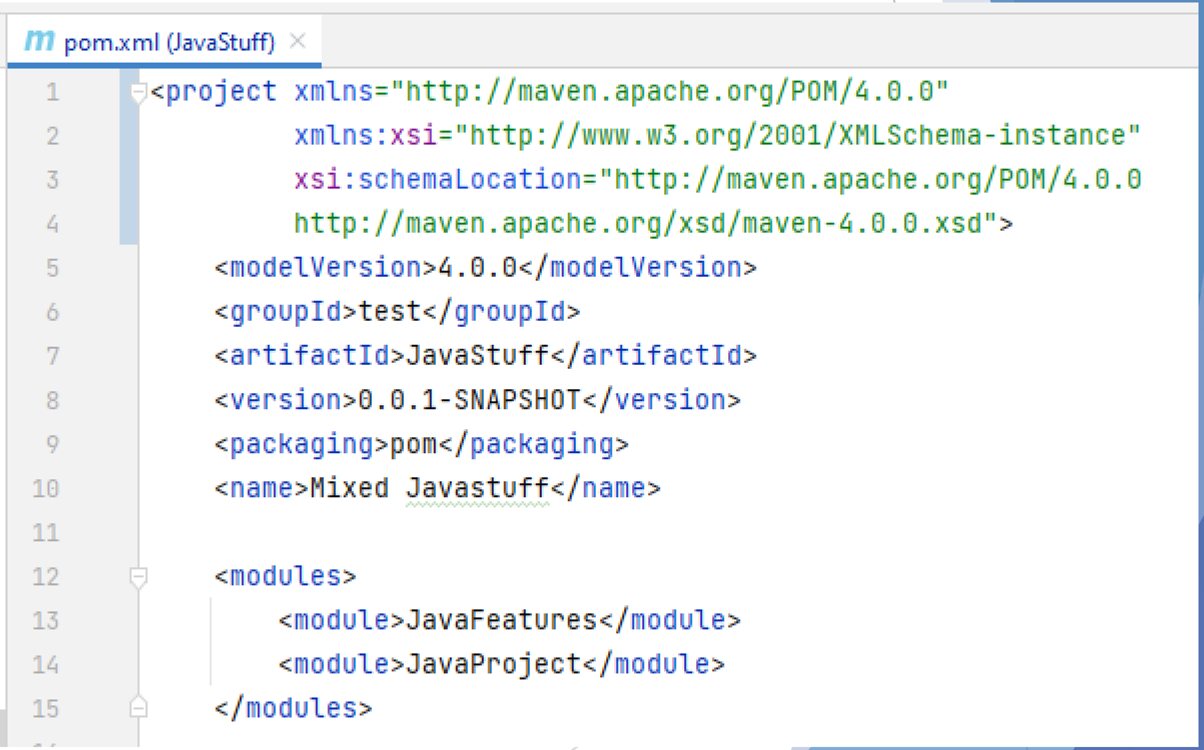
Motivation

Current state of Maven

- 20+ years old
- Consists of ~100 artifacts (plugins + components)
  - Not counting ~6.000 community plugins
- Maven 3.0 released 2010-10-08
  - Current version 3.9.8 from 2024-06-17

Maven`s POM (Model version 4.0.0)

- XML-Schema compatible since Maven 2 (2005)
- Build and usages information
  - Modules
  - Dependencies
  - Plugins (and their configuration)
  - Repositories, meta-data, etc.
- Deployed to Maven central



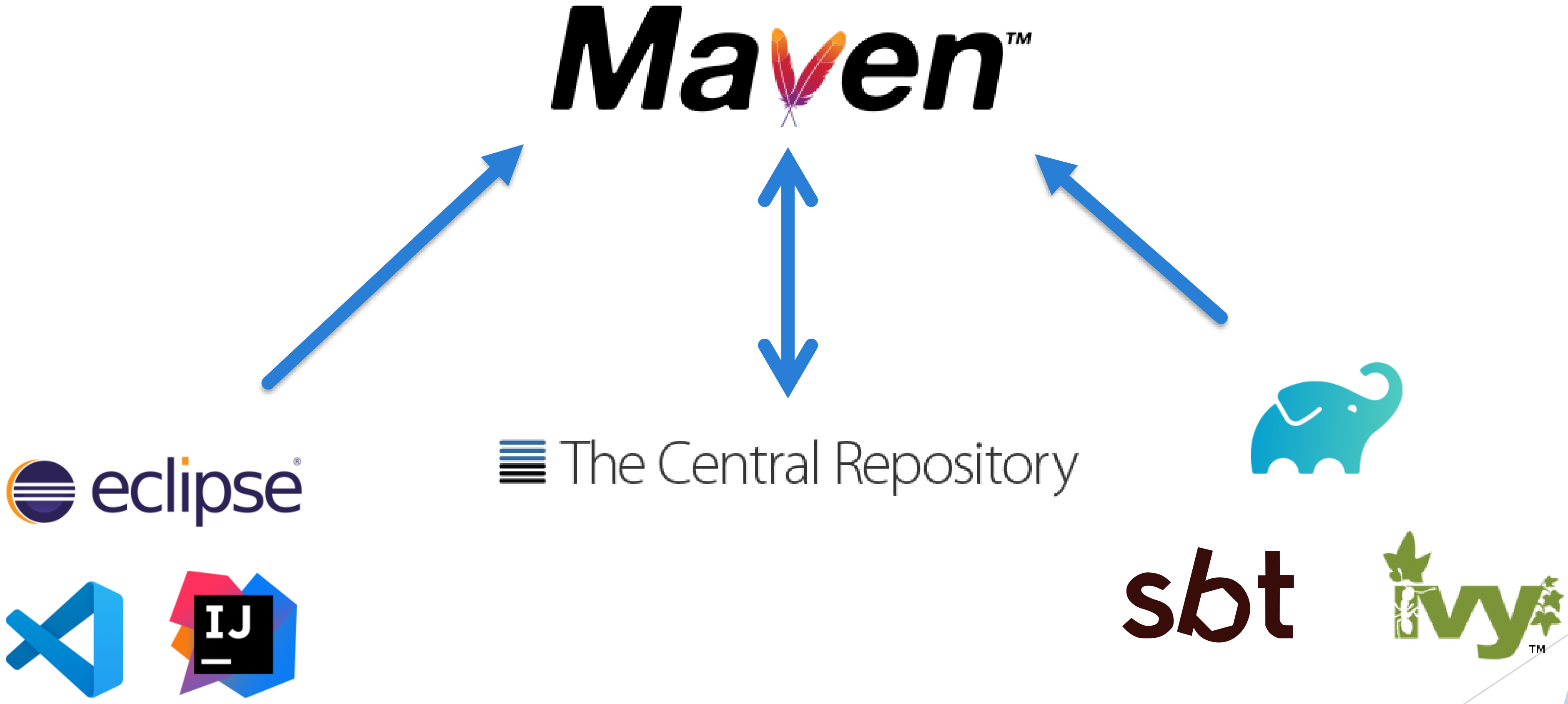
```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
         http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>test</groupId>
    <artifactId>JavaStuff</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>pom</packaging>
    <name>Mixed Javastuff</name>

    <modules>
        <module>JavaFeatures</module>
        <module>JavaProject</module>
    </modules>
```

Nobody is perfect

- POM v4 not very flexible, POM v5 is needed
- Maven 3 has some very old, redundant and problematic (e.g. mutable) code
  - Many plugins still use Maven 2 (and older) methods

More than a tool



> “With the Maven build schema preserved in amber, we can’t evolve much: we’ll stay forever with Maven 3 minor releases, unable to implement improvements that we imagine will require seriously updating the POM schema…”

Hervé Boutemy

https://www.javaadvent.com/2021/12/from-maven-3-to-maven-5.html

Building bridges

- Maven 5 could introduce POM v5
- Maven 4 will clear the way

Major changes in Maven 4

Disclaimer: Still beta (3) – things may change!

GREAT STORIES ALWAYS START
WITH AN AXE AND A BEER
#URBANWOODCUTTER

POM changes

- „Cleaving“ the POM 🪓
  - Build POM
  - Consumer POM
- Based on POM v4
  - Consumer POM with POM v4.0
  - Build POM with POM v4.1 🍺

Comparing POMs

	Build POM	Consumer POM *
POM version	4.1.0	4.0.0
Dependencies - Full parent / module - 3rd parties	 ❌ ✅	 ❌ ✅
Properties	✅	❌
Plugin configuration	✅	❌
Repository information	✅	✅
Profiles	✅	✅
Project information / Environment settings	✅	✅
Deployment to Remote	✅	✅

* Not for type „pom“

New packaging type: bom

- New feature in Maven 4 for BOM
- Requires Model version 4.1.0
- Automatically handles projects versions when creating a BOM file

- Example by Karl Heinz Marbaise (at IntelliJ IDEA Conf 2024)
  - https://www.youtube.com/watch?v=ZD_YxTmQ16Q&t=16710s

Required Java Version

- In March 2024: Vote passed for lifting Maven 4 to $\boxed{1}\boxed{7}$
  - First Java 17 version 4.0.0-alpha 13 (2024-03-06)

Overview: https://maven.apache.org/docs/history.html

Required Java Version

TO RUN MAVEN ⚠

(not your app)

Code updates

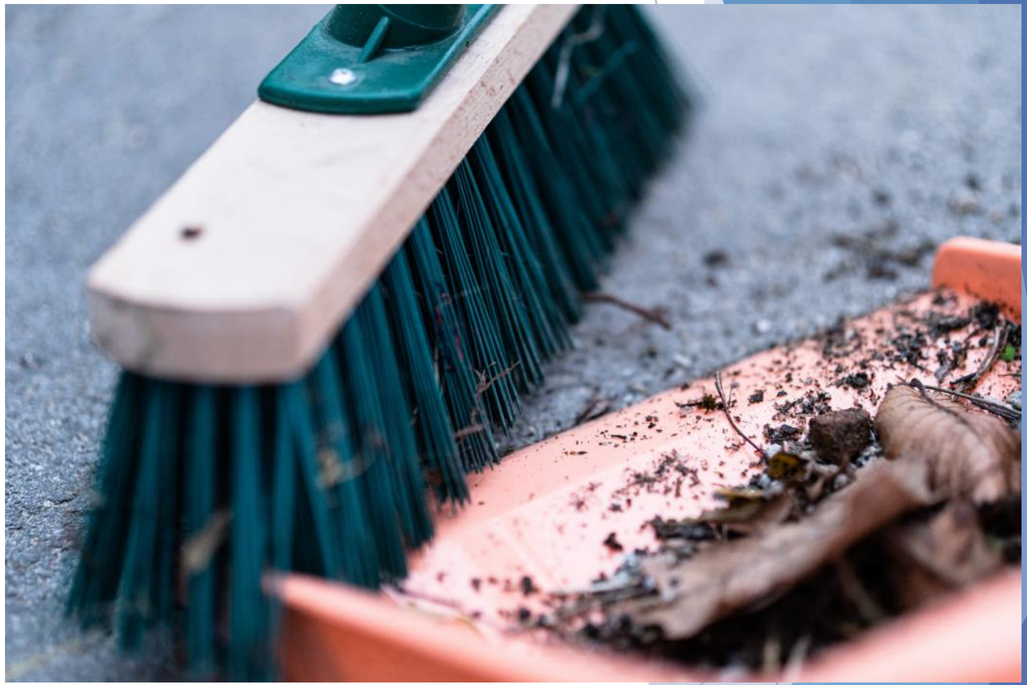
- Updated to Java 8 (will be 17)
  - Use new Java features / improvements
  - Will continue to 17 when baseline is set

- More multi-threading usages

- Updated default version of Maven plugins and dependencies (e.g. JUnit 5)

API cleanup

- Remove old Maven 1/2 [Plugin] APIs
  - Warnings in 3.9 / Failures in 4.0
- "Plexus" dependency resolver removed
  - Deprecated since Maven 3.2 (2010)



Improved user experience

Reactor improvements

- Resuming in multi-module projects **much** better
  - “—also-make” together with “—resume-from”
  - Also detecting modules build successful when building in parallel
- Aware of subfolder build
- Better support for multi-module which are multiple levels deep

Summary: https://maarten.mulders.it/2020/11/whats-new-in-maven-4/

„rootDirectory“ and „topDirectory“ variables

Name	Scope	Definition	Always?
`project.rootDirectory`	project	`.mvn`-folder or `root`-attribute in pom	No
`session.topDirectory`	session	current directory or `--file` argument	Yes
`session.rootDirectory`	session	`.mvn`-folder or `root`-attribute in pom for the topDirectory project	No

Summary:

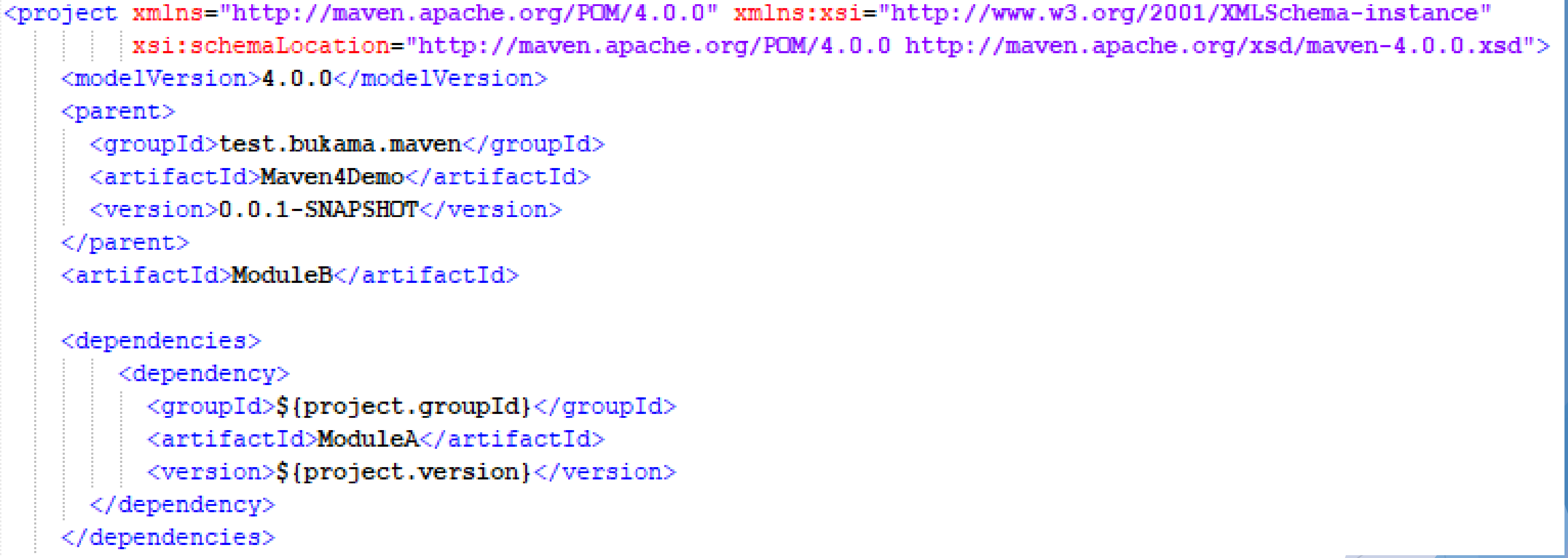
https://issues.apache.org/jira/browse/MNG-7038

https://github.com/apache/maven/pull/1061

Multi-module projects improvements

- Consistent timestamps for all module builds in a multi-module project
- Deploy all or none
- Automatic versioning 🥰
  - No parent version needed anymore
  - No versions for project-own modules needed anymore

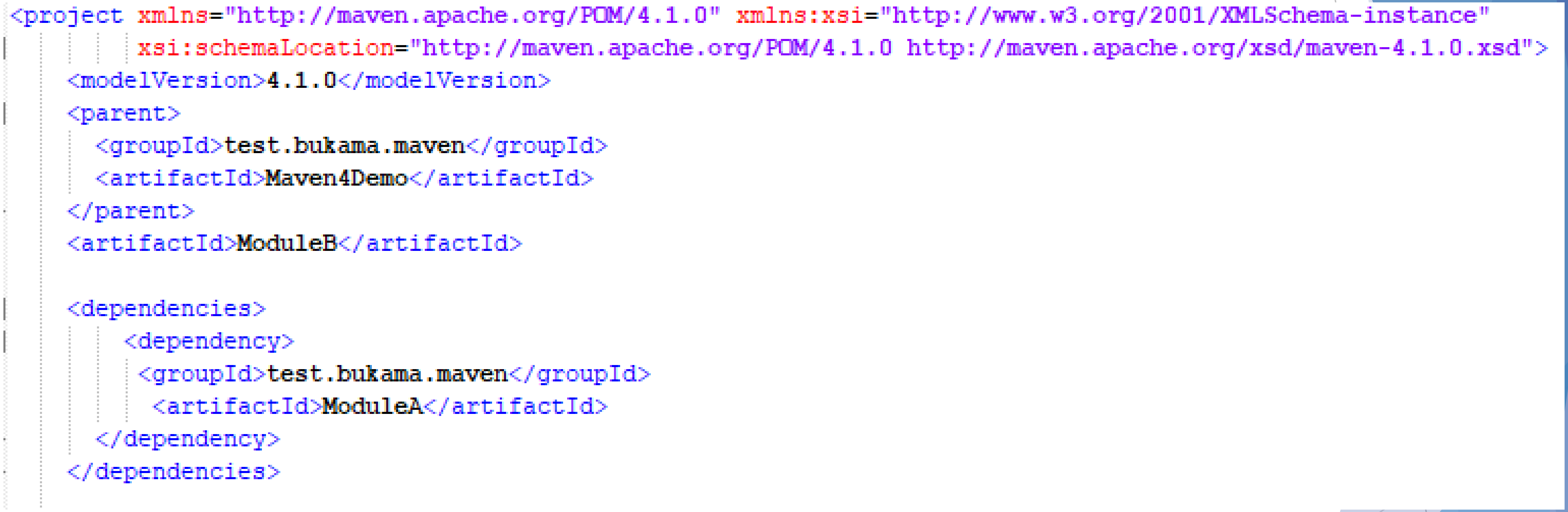
Multi-module project (Maven 3)



```xml
<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>
    <parent>
      <groupId>test.bukama.maven</groupId>
      <artifactId>Maven4Demo</artifactId>
      <version>0.0.1-SNAPSHOT</version>
    </parent>
    <artifactId>ModuleB</artifactId>

    <dependencies>
        <dependency>
          <groupId>${project.groupId}</groupId>
          <artifactId>ModuleA</artifactId>
          <version>${project.version}</version>
      </dependency>
    </dependencies>
```

Multi-module project (Maven 4)



```xml
<project xmlns="http://maven.apache.org/POM/4.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.1.0 http://maven.apache.org/xsd/maven-4.1.0.xsd">
    <modelVersion>4.1.0</modelVersion>
    <parent>
      <groupId>test.bukama.maven</groupId>
      <artifactId>Maven4Demo</artifactId>
    </parent>
    <artifactId>ModuleB</artifactId>

    <dependencies>
        <dependency>
         <groupId>test.bukama.maven</groupId>
          <artifactId>ModuleA</artifactId>
      </dependency>
    </dependencies>
```

Maven Wrapper

- Ensures the „right“ Maven version for the project

- Moved to Maven distribution (supports also Maven 3)

More information: https://maven.apache.org/wrapper/index.html

Build improvements

- „Fail on severity“-Parameter
  - Breaking builds on messages with given severity, e.g. „-fos WARN“
  - Improves project maintenance
  - Some of the current warnings will break your build in Maven 4!
- CI-friendly build variables (e.g. “${revision}”)
  - First approach in 3.5.0 (back then in combination with flatten-plugin)
  - https://blog.soebes.io/posts/2024/03/2024-03-31-maven-4-part-i/

Glimpse into the future

What might (or might not) be possible in the future?

⚠ DISCLAIMER ⚠

Imaginable POM changes

- Alternative syntaxes for POMs
  - https://issues.apache.org/jira/browse/MNG-7836
- Polyglot-POM
- Reduced dependency definition

Imaginable build changes

- Separate repositories for SNAPSHOT and public releases
  - Since 3.9: Multiple local repositories for different remote repositories

- Multiple / separated caches for different executions

- Deploying fully resolved dependency information

Recap: Maven 4

- Introduction Build & Consumer POM
- Minimum Java 17 with lots of code / API cleanup
- Multiple UX improvements
  - Reactor
  - Automatic versioning
  - Official variables for project root

What can/should one do?

- Update to latest Maven 3.9.x
- Prepare build environment to run Maven on Java 17
- Test out Maven 4 beta
- Give feedback to Apache Maven team

- 📧 users@maven.apache.org
- 🌍 https://maven.apache.org/

Images

- Freepik.com:
  - Wahrsagerin
- Unsplash.com
  - Warning Sign
  - Brown wooden bridge
  - Broom

- Pixabay.com
  Feedback
- Wikipedia.org
  - Apache Maven Icon
  - Insekt in Bernstein