CS544 EA

Overview

Intro: Maven

Maven

- We will use Maven for this course
  - **Industry standard** for serious Java Development
  - Build System (what IDEs also do)
  - Dependency Manager (downloads JARs)

- You can use Maven with any IDE
  - Some exercises may have examples from a specific IDE
  - You are free to use whatever you want

pom.xml

- **Project Object Model** (POM)
  - Tells Maven what your project is and needs

- 3 main areas:
  - Project Description
  - Dependencies
  - Plugins (special compiling / packaging)

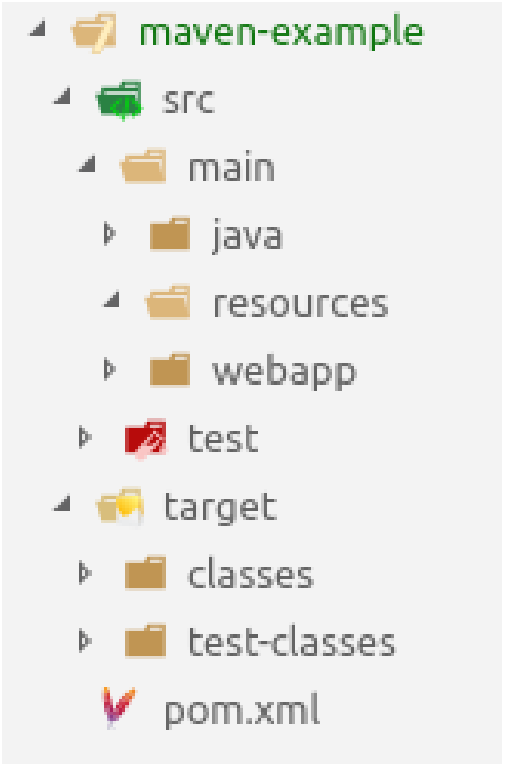
Creating a Project

- Maven has archetypes
  - Starter projects for certain things
  - Unfortunately **often very out of date**
- IDEs can often also generate Maven projects
  - Some use archetypes (out of date)
  - Some give you something decent
- You can also just copy or create your own

See: https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html

Directory Structure

- A Maven project contains:
  - **src** directory for source code
  - target directory (compiled / packaged code)
  - pom.xml
- The src directory contains:
  - **main** for program code and test for test code
- Main contains:
  - **java** for .java files (usually in packages / directories)
  - **resources** for non-java (config) files on the classpath
  - **webapp** for web related content



Sometimes the resources and webapp directories are not automatically generated.
You can always add them manually!

Example POM

```
<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>edu.mum.cs544</groupId>
      <artifactId>maven-example</artifactId>
      <version>0.0.1-SNAPSHOT</version>

      <dependencies>
            <dependency>
                  <groupId>org.springframework</groupId>
                  <artifactId>spring-context</artifactId>
                  <version>5.0.8.RELEASE</version>
            </dependency>
      </dependencies>

      <build>
            <plugins>
                  <plugin>
                        <groupId>org.codehaus.mojo</groupId>
                        <artifactId>exec-maven-plugin</artifactId>
                        <version>1.4.0</version>
                        <configuration>
                              <mainClass>edu.mum.cs544.App</mainClass>
                        </configuration>
                  </plugin>
            </plugins>
      </build>
</project>
```

Project Description

Optional Dependencies

Optional Build Plugins

Without an IDE

- Maven is a **command line** tool
  - Download from: http://maven.apache.org
  - Install on your PATH

- Inside a project directory type:

```
mvn package
mvn exec:java
```

Compiles your code and puts into JAR

If you have the exec-maven-plugin

Summary

- Maven is an industry standard build system
  - Downloads configured dependencies for you
  - Standardized directory structure
  - Builds, tests, and packages your code
  - Works with any Java IDE, or on its own