

Dependencies

Bryan Hansen

twitter: bh5k

<http://www.linkedin.com/in/hansenbryan>



Outline

- Versions
- Types
- Transitive Dependencies
- Scopes

Dependency

- **Just list the dependency that we want**
 - Transitive dependencies will be pulled in by Maven
- **Need at a minimum 3 things:**
 - groupId
 - artifactId
 - version

```
<dependencies>
  <dependency>
    <groupId>commons-lang</groupId>
    <artifactId>commons-lang</artifactId>
    <version>2.1</version>
  </dependency>
</dependencies>
```

Versions

- **Development starts off as a SNAPSHOT**
 - myapp-1.0-SNAPSHOT.jar
 - Changes always downloaded
 - Saves you from rereleasing versions for development
 - Never deploy to production with a SNAPSHOT
- **A release doesn't have a specific naming convention**
 - myapp-1.0.jar
 - myapp-1.0.1.jar
- **Industry common terms, but don't affect maven**
 - myapp-1.0-M1.jar (milestone release)
 - myapp-1.0-RC1.jar (release candidate)
 - myapp-1.0-RELEASE.jar (release)
 - myapp-1.0-Final.jar (release)

Types

- **Current core packaging types are:**
 - pom, jar, maven-plugin, ejb, war, ear, rar, par
 - The default packaging type is jar
- **The type of pom is referred to as a dependency pom**
 - Downloads dependencies from that pom

```
<groupId>com.pluralsight</groupId>  
<artifactId>HelloWorld</artifactId>  
<version>1.0-SNAPSHOT</version>  
<modelVersion>4.0.0</modelVersion>  
<packaging>jar</packaging>
```

Transitive Dependencies

- The main reason people begin using maven
- If we add a dependency:

```
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>4.1.6.Final</version>
  <scope>compile</scope>
</dependency>
```

- If we add a dependency it downloads it's transitive dependencies:

```
antlr : 2.7.7 [compile]
dom4j : 1.6.1 [compile]
hibernate-commons-annotations : 4.0.1.Final [compile]
hibernate-core : 4.1.6.Final [compile]
hibernate-jpa-2.0-api : 1.0.1.Final [compile]
javassist : 3.15.0-GA [compile]
jboss-logging : 3.1.0.GA [compile]
jboss-transaction-api_1.1_spec : 1.0.0.Final [compile]
```

Scopes

- **There are 6 scopes available for dependencies:**
 - compile – default scope, artifacts available everywhere
 - provided – like compile, means that the artifact is going to be provided where it is deployed
 - servlet-api.jar
 - xml-apis
 - Available in all phases, but not included in final artifact
 - runtime – not needed for compilation, but needed for execution
 - Not available for compilation, but included in all other phases
 - Not included in final artifact
 - test – only available for the test compilation and execution phase
 - system – similar to provided, you specify a path to the jar on your file system
 - Very brittle and defeats the purpose of maven, don't use!
 - import – advanced topic, deals with dependencyManagement sections

Demo

- **Let's look at what the pom.xml will look like when we:**
 - Add a dependency
 - Reference transitive dependencies
 - Use scopes

Summary

- **Version numbers are up to your corporate strategy**
 - SNAPSHOT has a unique meaning
- **Supported types and what they mean to your application**
- **Transitive dependencies and how they are downloaded**
- **Scopes allow us to compile or test our code, but not include artifacts that shouldn't be in the packaged code**