

Spring & Maven

Sang Shin
JPassion.com
“Learn with Passion!”



Topics

- Maven Repositories for Spring artifacts
- Creating Spring Maven projects

Maven Repositories for Spring Artifacts

Maven Repositories

- Spring publishes its artifacts to two different places:
 - > Maven Central
 - > Enterprise Bundle Repository (EBR)
- Maven Central
 - > Default repository
 - > Does not require any special configuration to use
- Enterprise Bundle Repository (EBR)
 - > Run by SpringSource
 - > Hosts OSGi-ready versions of hundreds of open source enterprise libraries that are commonly used when developing Spring applications

Which Repository to use?

- In general, if you care about OSGi, use the EBR, since it houses OSGi compatible artifacts for all of Spring's dependencies, such as Hibernate and Freemarker
 - > If OSGi does not matter to you, either place works
- In general, pick one place or the other for your project; do not mix them
 - > This is particularly important since EBR artifacts use a different naming convention than Maven Central artifacts.

pom.xml, which uses Maven Central

```
<!-- Shared version number properties -->
<properties>
  <org.springframework.version>3.0.5.RELEASE</org.springframework.version>
</properties>

<!--
  Core utilities used by other modules.
  Define this if you use Spring Utility APIs (org.springframework.core.*/org.springframework.util.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>${org.springframework.version}</version>
</dependency>

<!--
  Expression Language (depends on spring-core)
  Define this if you use Spring Expression APIs (org.springframework.expression.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-expression</artifactId>
  <version>${org.springframework.version}</version>
</dependency>

<!-- more -->
```

pom.xml, which uses EBR

```
<!-- Shared version number properties -->
<properties>
  <org.springframework.version>3.0.0.RELEASE</org.springframework.version>
</properties>

<!--
  Core utilities used by other modules.
  Define this if you use Spring Utility APIs (org.springframework.core.*/org.springframework.util.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>org.springframework.core</artifactId>
  <version>${org.springframework.version}</version>
</dependency>

<!--
  Expression Language (depends on core)
  Define this if you use Spring Expression APIs (org.springframework.expression.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>org.springframework.expression</artifactId>
  <version>${org.springframework.version}</version>
</dependency>
```

pom.xml, which uses EBR

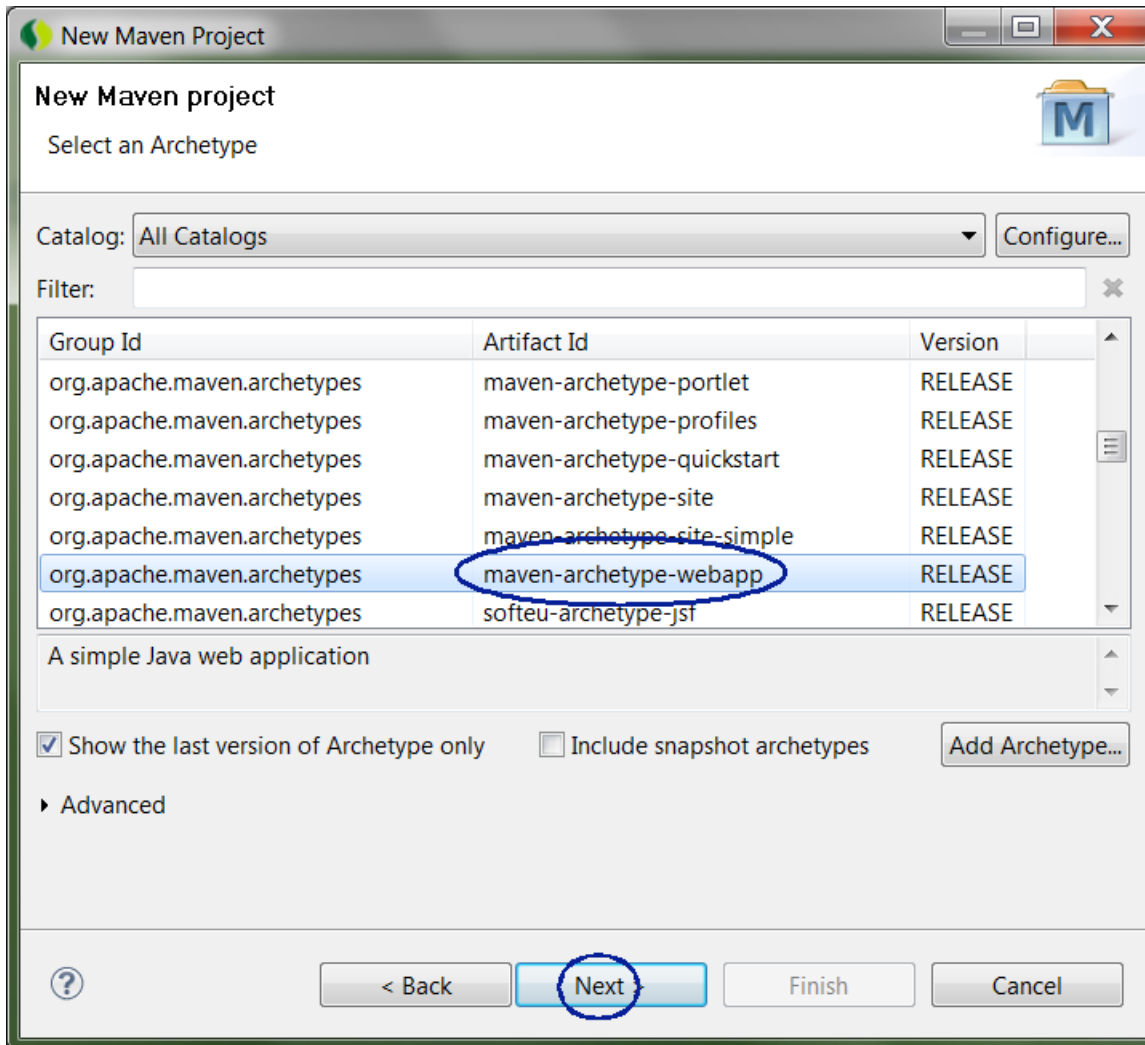
```
<repository>
  <id>com.springsource.repository.bundles.release</id>
  <name>EBR Spring Release Repository</name>
  <url>http:// repository.springsource.com/maven/bundles/release</url>
</repository>
<repository>
  <id>com.springsource.repository.bundles.external</id>
  <name>EBR External Release Repository</name>
  <url>http:// repository.springsource.com/maven/bundles/external</url>
</repository>
```

Creating Spring Maven Projects

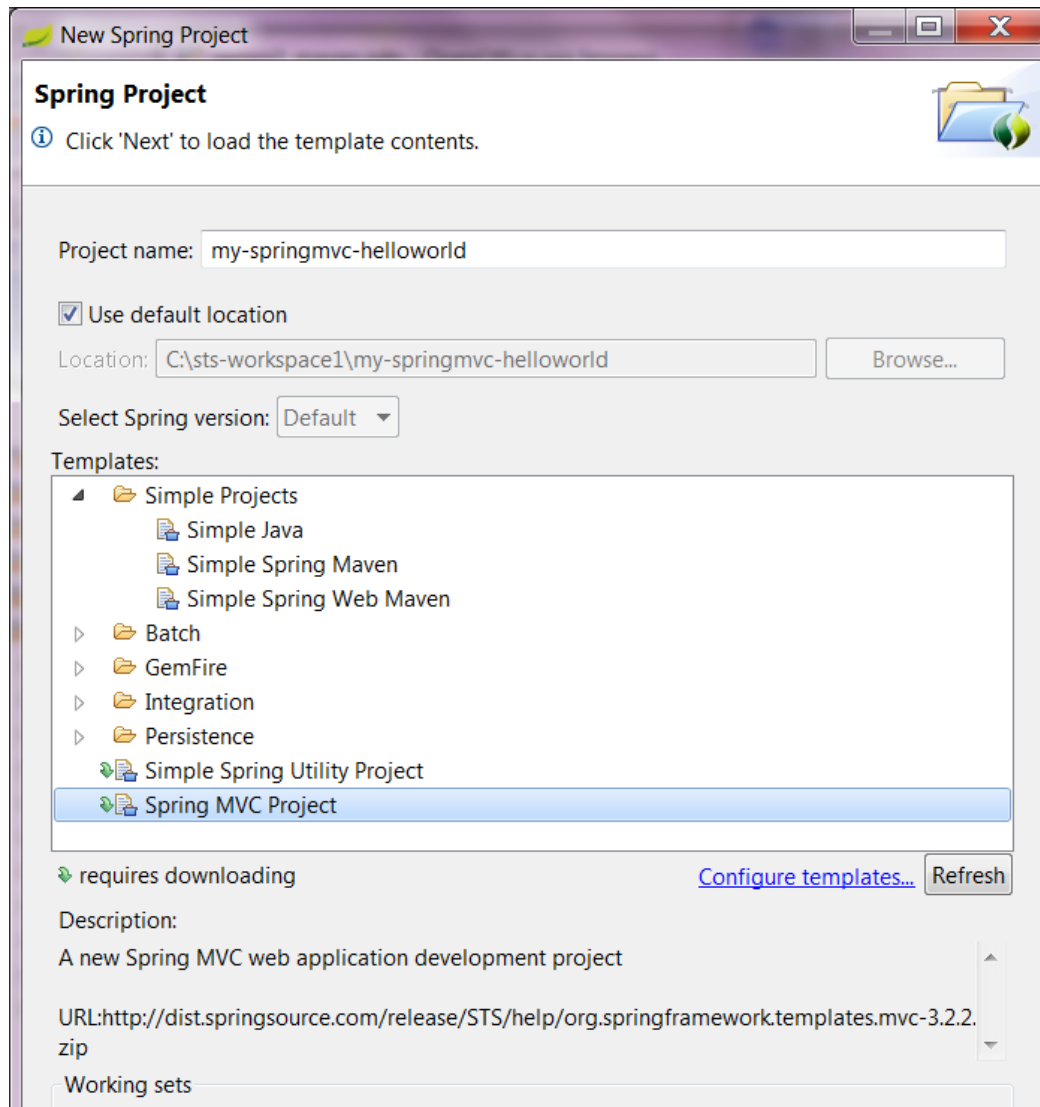
Options of Creating SpringMVC Maven Projects

- Option #1
 - > Create Maven Web app archetype and add SpringMVC code
- Option #2
 - > Create Maven project from Spring STS template project if you are using Spring STS or Eclipse+STS plug-in
- Option #3
 - > Copy existing SpringMVC Maven project and add/modify

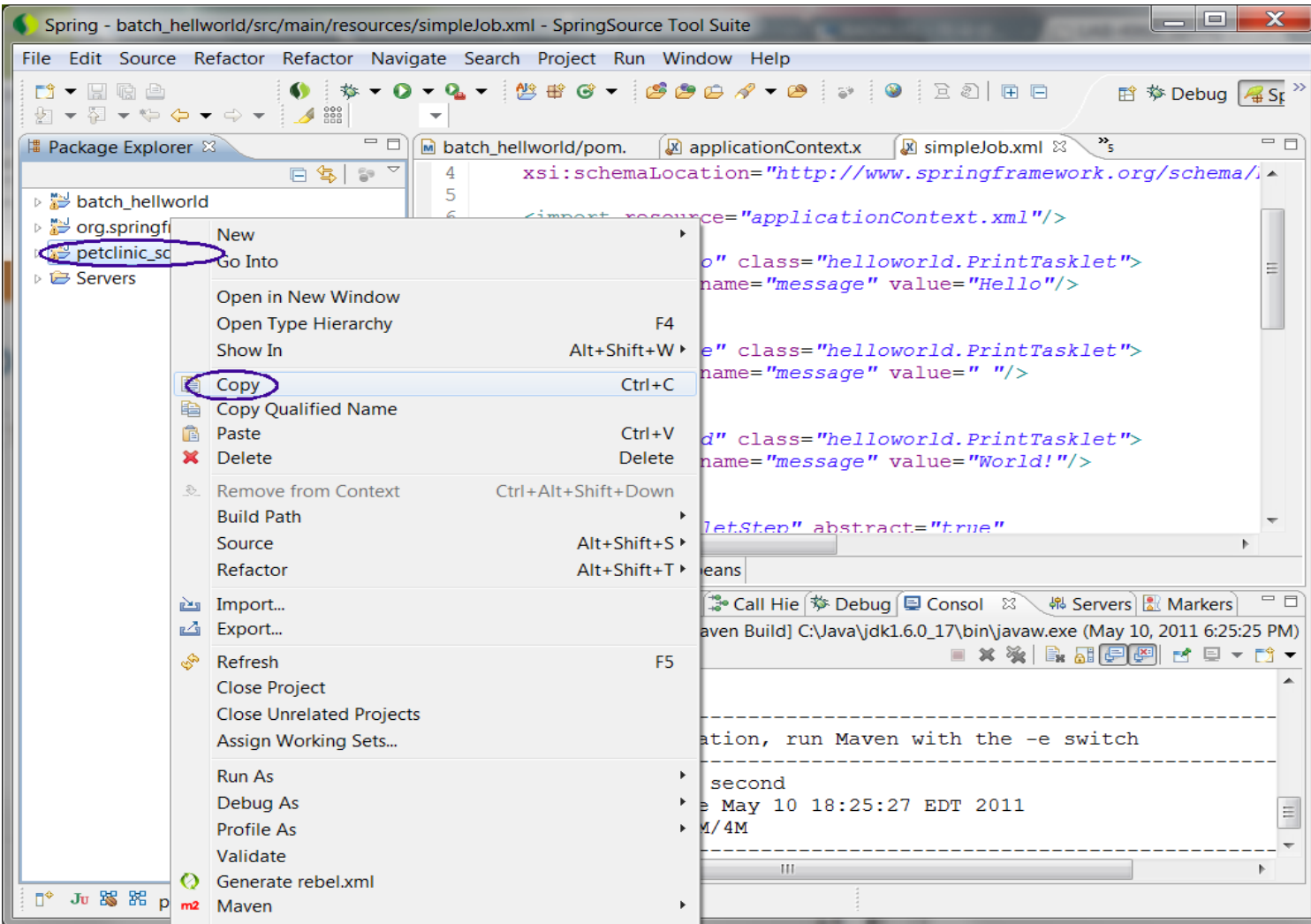
Option #1: maven-archetype-webapp



Option #2: Spring MVC Template



Option #3: Copy Spring MVC project



Lab:

Exercise 1-5

4969_spring3_maven.zip



Learn with Passion!
JPassion.com

