Top of Form

Creating Maven multi-module project with Eclipse

**Created on:**August 17, 2014 | **Last updated on:**November 26, 2017  [websystiqueadmin](http://websystique.com/author/blogadminuser/)

In this post, we will see how to create a maven Multi-module project. Multi-module project structuring can be handy when we have to share a particular module between different projects. For instance, you can have different web-applications which are using same module for interacting with database.

We will develop a parent-project and three sub-modules projects (one model library and two web applications), where both web-applications will share the same model library.

**Step 1 : Create Parent project**

|  |
| --- |
| mvn archetype:generate -DgroupId=com.websystique.multimodule -DartifactId=parent-project |

This command will create a new project with name ‘parent-project’ in your workspace.

**Step 2 : Update pom.xml to declare it as parent project**

Open pom.xml of above created parent-project and change the packaging to ‘pom’.

|  |
| --- |
| <packaging>pom</packaging> |

/parent-project/pom.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>com.websystique.multimodule</groupId>    <artifactId>parent-project</artifactId>    <version>1.0-SNAPSHOT</version>    <packaging>pom</packaging>      <name>parent-project</name>    <url>[http://maven.apache.org](http://maven.apache.org/)</url>      <properties>      <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>    </properties>      <dependencies>      <dependency>        <groupId>junit</groupId>        <artifactId>junit</artifactId>        <version>3.8.1</version>        <scope>test</scope>      </dependency>    </dependencies>  </project> |

**Step 3 : Create sub-modules**

Via command line, navigate to the directory of parent-project, and fire following commands

|  |
| --- |
| cd parent-project  mvn archetype:generate -DgroupId=com.websystique.multimodule  -DartifactId=model-lib  mvn archetype:generate -DgroupId=com.websystique.multimodule  -DartifactId=webapp1  mvn archetype:generate -DgroupId=com.websystique.multimodule  -DartifactId=webapp2 |

Now if you open the parent-project pom.xml, you will find all three modules being added in there.

|  |
| --- |
| <modules>    <module>model-lib</module>    <module>webapp1</module>    <module>webapp2</module>  </modules> |

Also, in each sub-module’s pom.xml, a parent section being added.

|  |
| --- |
| <parent>    <groupId>com.websystique.multimodule</groupId>    <artifactId>parent-project</artifactId>    <version>1.0-SNAPSHOT</version>  </parent> |

**Step 4 : Update sub-modules pom.xml to produce appropriate output**

In our case, model-lib is a libreary(jar), we will set its’s packaging to jar.

|  |
| --- |
| <packaging>jar</packaging> |

On the other hand, webapp1 & webapp2 are intended to be WAR, so we need to changed their pom.xml to set the packaging to war.

|  |
| --- |
| <packaging>war</packaging> |

Below are the final pom.xml of each sub-module and parent-project

/parent-project/model-lib/pom.xml

|  |
| --- |
| <?xml version="1.0"?>  <project xsi:schemaLocation="<http://maven.apache.org/POM/4.0.0> <http://maven.apache.org/xsd/maven-4.0.0.xsd>" xmlns="<http://maven.apache.org/POM/4.0.0>"      xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>">    <modelVersion>4.0.0</modelVersion>    <parent>      <groupId>com.websystique.multimodule</groupId>      <artifactId>parent-project</artifactId>      <version>1.0-SNAPSHOT</version>    </parent>    <groupId>com.websystique.multimodule</groupId>    <artifactId>model-lib</artifactId>    <version>1.0-SNAPSHOT</version>    <name>model-lib</name>    <packaging>jar</packaging>    <url>[http://maven.apache.org](http://maven.apache.org/)</url>    <properties>      <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>    </properties>    <dependencies>      <dependency>        <groupId>junit</groupId>        <artifactId>junit</artifactId>        <version>3.8.1</version>        <scope>test</scope>      </dependency>    </dependencies>  </project> |

/parent-project/webapp1/pom.xml

|  |
| --- |
| <?xml version="1.0"?>  <project xsi:schemaLocation="<http://maven.apache.org/POM/4.0.0> <http://maven.apache.org/xsd/maven-4.0.0.xsd>" xmlns="<http://maven.apache.org/POM/4.0.0>"      xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>">    <modelVersion>4.0.0</modelVersion>    <parent>      <groupId>com.websystique.multimodule</groupId>      <artifactId>parent-project</artifactId>      <version>1.0-SNAPSHOT</version>    </parent>    <groupId>com.websystique.multimodule</groupId>    <artifactId>webapp1</artifactId>    <version>1.0-SNAPSHOT</version>    <name>webapp1</name>    <packaging>war</packaging>    <url>[http://maven.apache.org](http://maven.apache.org/)</url>    <properties>      <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>    </properties>    <dependencies>      <dependency>        <groupId>junit</groupId>        <artifactId>junit</artifactId>        <version>3.8.1</version>        <scope>test</scope>      </dependency>    </dependencies>  </project> |

/parent-project/webapp2/pom.xml

|  |
| --- |
| <?xml version="1.0"?>  <project xsi:schemaLocation="<http://maven.apache.org/POM/4.0.0> <http://maven.apache.org/xsd/maven-4.0.0.xsd>" xmlns="<http://maven.apache.org/POM/4.0.0>"      xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>">    <modelVersion>4.0.0</modelVersion>    <parent>      <groupId>com.websystique.multimodule</groupId>      <artifactId>parent-project</artifactId>      <version>1.0-SNAPSHOT</version>    </parent>    <groupId>com.websystique.multimodule</groupId>    <artifactId>webapp2</artifactId>    <version>1.0-SNAPSHOT</version>    <name>webapp2</name>    <packaging>war</packaging>    <url>[http://maven.apache.org](http://maven.apache.org/)</url>    <properties>      <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>    </properties>    <dependencies>      <dependency>        <groupId>junit</groupId>        <artifactId>junit</artifactId>        <version>3.8.1</version>        <scope>test</scope>      </dependency>    </dependencies>  </project> |

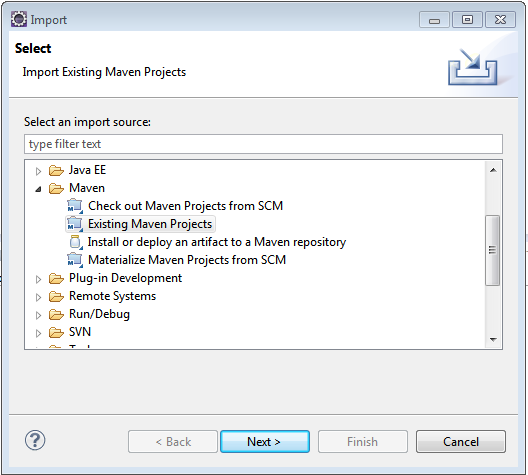
And the parent-project pom.xml is

/parent-project/pom.xml

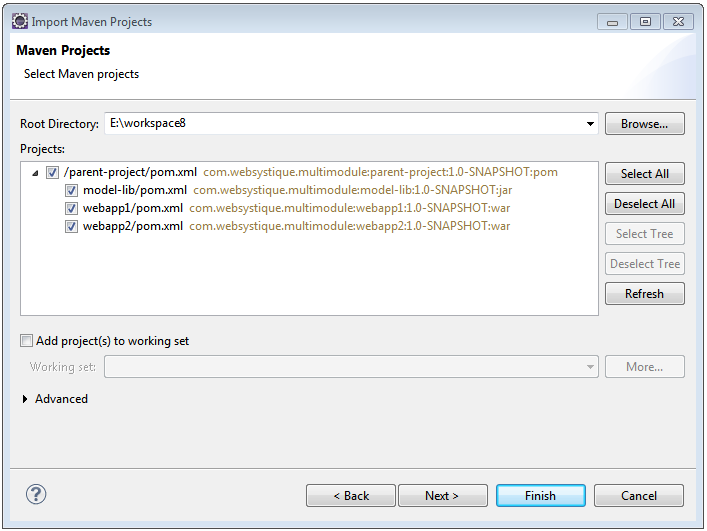
|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <project xmlns="<http://maven.apache.org/POM/4.0.0>" xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>" xsi:schemaLocation="<http://maven.apache.org/POM/4.0.0> <http://maven.apache.org/xsd/maven-4.0.0.xsd>">    <modelVersion>4.0.0</modelVersion>      <groupId>com.websystique.multimodule</groupId>    <artifactId>parent-project</artifactId>    <version>1.0-SNAPSHOT</version>    <packaging>pom</packaging>      <name>parent-project</name>    <url>[http://maven.apache.org](http://maven.apache.org/)</url>      <properties>      <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>    </properties>      <dependencies>      <dependency>        <groupId>junit</groupId>        <artifactId>junit</artifactId>        <version>3.8.1</version>        <scope>test</scope>      </dependency>    </dependencies>    <modules>      <module>model-lib</module>      <module>webapp1</module>      <module>webapp2</module>    </modules>  </project> |

**Step 5 : Import these projects as ‘Existing maven projects’ in Eclipse**

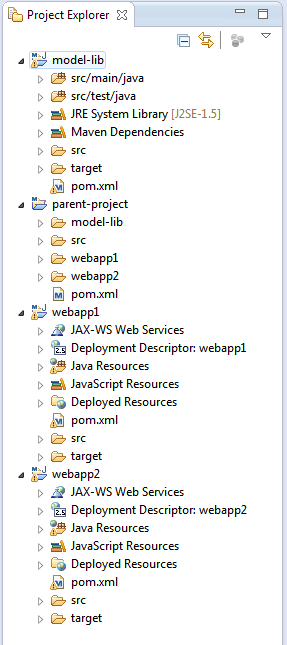
In Eclipse , File->Import->Maven->Existing maven projects



click next, select your workspace as root directory , you should see something similar :

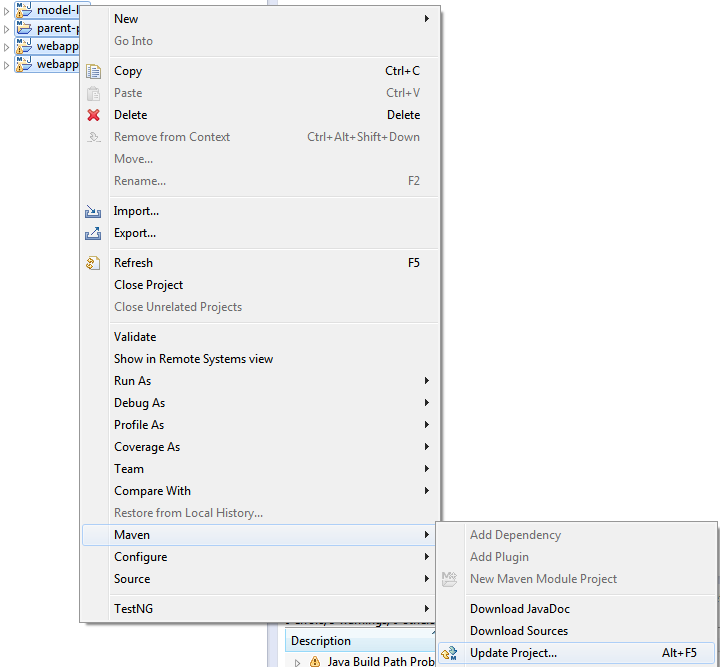


click finish, you should see the projects got imported successfully in Eclipse.



Now when you build the parent-project, all your child project will be build successively.

One remark: Although it’s not mandatory, when you change any of your project, it’s advisable to perform a Maven-> Update project do get the changes reflected in all the projects which depends on this project.



That’s it.

**References**

* [Maven Pom introduction](http://maven.apache.org/guides/introduction/introduction-to-the-pom.html)
* [Maven life cycle](http://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html)
* [MvnRepository](http://mvnrepository.com/)



[websystiqueadmin](http://websystique.com/author/blogadminuser/)

If you like tutorials on this site, why not take a step further and connect me on [Facebook](https://www.facebook.com/websystique) , [Google Plus](https://plus.google.com/+WebsystiqueDotCom/posts) & [Twitter](https://twitter.com/Websystique) as well? I would love to hear your thoughts on these articles, it will help me improve further our learning process.

If you appreciate the effort I have put in this learning site, help me improve the visibility of this site towards global audience by sharing and linking this site from within and beyond your network. You & your friends can always link my site from your site on [www.websystique.com](http://www.websystique.com/), and share the learning.

After all, we are here to learn together, aren’t we?

**Related Posts:**

1. [**Create a Maven Web Project with Eclipse**](http://websystique.com/maven/create-a-maven-web-project-with-eclipse/)
2. [**AngularJS Modules Explained**](http://websystique.com/angularjs/angularjs-modules-explained/)
3. [**Create a Maven Project with CommandLine**](http://websystique.com/maven/create-a-maven-project-with-commandline/)
4. [**Angularjs Routing Tutorial using ui-router**](http://websystique.com/angularjs/angularjs-routing-tutorial-using-ui-router/)

[maven](http://websystique.com/category/maven/).  [permalink](http://websystique.com/maven/creating-maven-multi-module-project-with-eclipse/).

Post navigation

[← Spring 4 MVC ContentNegotiatingViewResolver example](http://websystique.com/springmvc/spring-4-mvc-contentnegotiatingviewresolver-example/)

[Spring 4 Hello World Example →](http://websystique.com/spring/spring-4-hello-world-example-annotation-tutorial-full-example/)

* Mohammed Sirajuddin

mvn archetype:generate -DgroupId=com.websystique.multimodule -DartifactId=parent-project

* + Mohammed Sirajuddin

create goal is deprecated please update your post.

* Pingback: [Eclipse JEE : Passer à Maven | Epic Coffee Mix'](https://blog.infovergne.com/eclipse-jee-passer-a-maven/)()
* Sudhamohan Panda

very good step by step example

* manjush b

Its a good example , How can I configure spring boot with this example, I tried but didnt work.  
I tried with this example <http://websystique.com/spring-boot/spring-boot-angularjs-spring-data-jpa-crud-app-example/>  
Please inform if you have any article related to maven multi module and spring boot

* farhan

You have given maven command as mvn archetype:create but It did not work until I used mvn archetype:generate.

* + websystique

Thanks Farhan, i’ve updated the post with correct type.Thanks again.

* + - Sangimed

Hi, actually you didn’t update the maven command that creates the parent project (mvn archetype:create -DgroupId=com.websystique.multimodule -DartifactId=parent-project) so you need to replace create with generate in this one too :-)

* + - * Cor

Yes I had the same problem. I thank you not for updating the post at step 1.

* Dhyanandra Singh

how to configure springMVC, hibernate and JAx-Rs based maven multimodile project. That means how to configure appIntializer, appconfig in such a way that it will execute all sub project.

* websystique

test

Recent Posts

* [Spring Boot + AngularJS + Spring Data + JPA CRUD App Example](http://websystique.com/spring-boot/spring-boot-angularjs-spring-data-jpa-crud-app-example/)
* [Spring Boot Rest API Example](http://websystique.com/spring-boot/spring-boot-rest-api-example/)
* [Spring Boot WAR deployment example](http://websystique.com/spring-boot/spring-boot-war-deployment-example/)
* [Spring Boot Introduction + hello world example](http://websystique.com/spring-boot/spring-boot-introduction-hello-world-example/)
* [Secure Spring REST API using OAuth2](http://websystique.com/spring-security/secure-spring-rest-api-using-oauth2/)

Copyright © 2014-2017 WebSystique.com. All rights reserved.

**We use cookies to personalize content and ads, to provide the best browsing experience possible, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners. By continuing to use the site, you agree to the use of cookies.**[**more information**](http://websystique.com/maven/creating-maven-multi-module-project-with-eclipse/)**Accept**