

How To Create A Web Application Project With Maven



Posted on December 21, 2012 , Last modified : January 1, 2013

 By [mkyong](#)

In this tutorial, we will show you how to use Maven to create a Java web application (with Spring MVC) project, and make it support Eclipse IDE.

Tools used :

1. Maven 3.0.5
2. Eclipse 4.2
3. JDK 6
4. Spring 3.2.0.RELEASED
5. Tomcat 7

1. Web Application Project from Maven Template

In a terminal (*nix or Mac) or command prompt (Windows), navigate to the folder you want to store the project. Issue following command :

```
mvn archetype:generate -DgroupId={project-packaging} -DartifactId={project-name} -DarchetypeArtifactId=maven-archetype-webapp
```

This tell Maven to create a Java web application project from "maven-archetype-webapp" template.

For example,

```
$ pwd
/Users/mkyong/Documents/workspace

$ mvn archetype:generate -DgroupId=com.mkyong -DartifactId=CounterWebApp -DarchetypeArtifactId=maven-archetype-webapp
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----
[INFO]
[INFO] Generating project in Batch mode
[INFO] -----
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-webapp
[INFO] -----
[INFO] Parameter: groupId, Value: com.mkyong
[INFO] Parameter: packageName, Value: com.mkyong
[INFO] Parameter: package, Value: com.mkyong
[INFO] Parameter: artifactId, Value: CounterWebApp
[INFO] Parameter: basedir, Value: /Users/mkyong/Documents/workspace
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: /Users/mkyong/Documents/workspace/CounterWebApp
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.147s
[INFO] Finished at: Thu Dec 20 20:35:19 MYT 2012
[INFO] Final Memory: 12M/128M
[INFO] -----
```

In above case, a new web application project named "CounterWebApp", and the entire project directory structure is created automatically.



Learn Hadoop Programming

e!www.edureka.in

Online Training from Experts 24x7 Online Support



AdChoices

2. Maven Directory Layout

Maven created following web application directory layout. A standard deployment descriptor `web.xml` and Maven `pom.xml` are created.

```
CounterWebApp
|--src
|---main
|----resources
|----webapp
|-----index.jsp
|-----WEB-INF
|-----web.xml
|---pom.xml
```

Note

Please check this [official guide](#) to understand more.

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/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mkymong</groupId>
  <artifactId>CounterWebApp</artifactId>
  <packaging>war</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>CounterWebApp Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <finalName>CounterWebApp</finalName>
  </build>
</project>
```

web.xml – Servlet 2.3 is too old, consider upgrade to 2.5

```
<!DOCTYPE web-app PUBLIC
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>
  <display-name>Archetype Created Web Application</display-name>
</web-app>
```

index.jsp – A simple hello world html file

```
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
```

3. Eclipse IDE Support

To convert the Maven web project to support Eclipse IDE, in terminal, navigate to "CounterWebApp" folder, issue this command :

SpreadsheetGear
Performance Spreadsheet Components

NEW! WPF Controls

NEW! Silverlight Controls

NEW! Multithreaded Calculation Engine

NEW! 64 Additional Excel Functions

SpreadsheetGear
Download Free Trial

50% OFF
on Sarees & Kurtas

Visit Now

```
mvn eclipse:eclipse -Dwtpversion=2.0
```

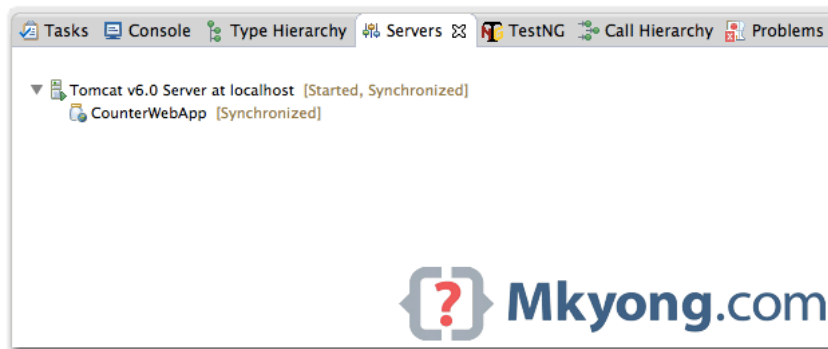
You must add the `-Dwtpversion=2.0` argument to make it as a Eclipse web project. Imports it into Eclipse IDE, a globe icon on top of project, means this is a web project in Eclipse.



You want Eclipse web project NOT Eclipse Java project.

Many users are confused, again, if you just issue `mvn eclipse:eclipse`, it will only convert the project as Eclipse Java project, add extra `-Dwtpversion=2.0` argument to make it as Eclipse web project.

Done. This web project is ready to deploy. Attached to Eclipse's Tomcat server plugin, and start it.



You can access the hello world jsp via – <http://localhost:8080/CounterWebApp/>

4. Update POM

To make above Maven web project to support Spring MVC framework, we need to touch up on the existing `pom.xml` :

1. Add compiler plugin to specify JDK6 to compile this project (default is using JDK1.4).
2. Add Spring frameworks dependencies.
3. Update junit to latest 4.11.

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/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mkyong</groupId>
  <artifactId>CounterWebApp</artifactId>
  <packaging>war</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>CounterWebApp Maven Webapp</name>
  <url>http://maven.apache.org</url>

  <properties>
    <spring.version>3.0.5.RELEASE</spring.version>
    <junit.version>4.11</junit.version>
    <jdk.version>1.6</jdk.version>
  </properties>

  <dependencies>
```

```

<!-- Spring 3 dependencies -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-core</artifactId>
    <version>${spring.version}</version>
</dependency>

<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-web</artifactId>
    <version>${spring.version}</version>
</dependency>

<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>${spring.version}</version>
</dependency>

<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>${junit.version}</version>
    <scope>test</scope>
</dependency>
</dependencies>
<build>
    <finalName>CounterWebApp</finalName>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.0</version>
            <configuration>
                <source>${jdk.version}</source>
                <target>${jdk.version}</target>
            </configuration>
        </plugin>
    </plugins>
</build>
</project>

```

5. Spring MVC REST

Create a Spring MVC controller class, with two simple methods to print a message.

```
/src/main/java/com/mkyong/controller/BaseController.java
```

```

package com.mkyong.controller;

import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

@Controller
@RequestMapping("/")
public class BaseController {

    @RequestMapping(value="/welcome", method = RequestMethod.GET)
    public String welcome(ModelMap model) {

        model.addAttribute("message", "Maven Web Project + Spring 3 MVC - welcome()");

        //Spring uses InternalResourceViewResolver and return back index.jsp
        return "index";

    }

    @RequestMapping(value="/welcome/{name}", method = RequestMethod.GET)
    public String welcomeName(@PathVariable String name, ModelMap model) {

        model.addAttribute("message", "Maven Web Project + Spring 3 MVC - " + name);
        return "index";

    }

}

```

Create a Spring configuration file, defines the Spring view resolver.

/src/main/webapp/WEB-INF/mvc-dispatcher-servlet.xml

```
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
         http://www.springframework.org/schema/beans
         http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
         http://www.springframework.org/schema/context
         http://www.springframework.org/schema/context/spring-context-3.0.xsd">

    <context:component-scan base-package="com.mkyong.controller" />

    <bean
        class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix">
            <value>/WEB-INF/pages/</value>
        </property>
        <property name="suffix">
            <value>.jsp</value>
        </property>
    </bean>

</beans>
```

Update existing `web.xml` to support Servlet 2.5 (the default Servlet 2.3 is too old), and also integrates Spring framework into this web application project via Spring's listener `ContextLoaderListener`.

/src/main/webapp/WEB-INF/web.xml

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
         http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
         version="2.5">

    <display-name>Counter Web Application</display-name>

    <servlet>
        <servlet-name>mvc-dispatcher</servlet-name>
        <servlet-class>
            org.springframework.web.servlet.DispatcherServlet
        </servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>mvc-dispatcher</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>

    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>/WEB-INF/mvc-dispatcher-servlet.xml</param-value>
    </context-param>

    <listener>
        <listener-class>
            org.springframework.web.context.ContextLoaderListener
        </listener-class>
    </listener>

</web-app>
```

Move existing `index.jsp` inside folder `WEB-INF`, to protect user access it directly. In additional, edit the file to print out the `${message}` variable that pass by the controller.

/src/main/webapp/WEB-INF/pages/index.jsp

```
<html>
<body>
<h2>Hello World!</h2>

<h4>Message : ${message}</h1>
</body>
</html>
```

Review the final directory structure

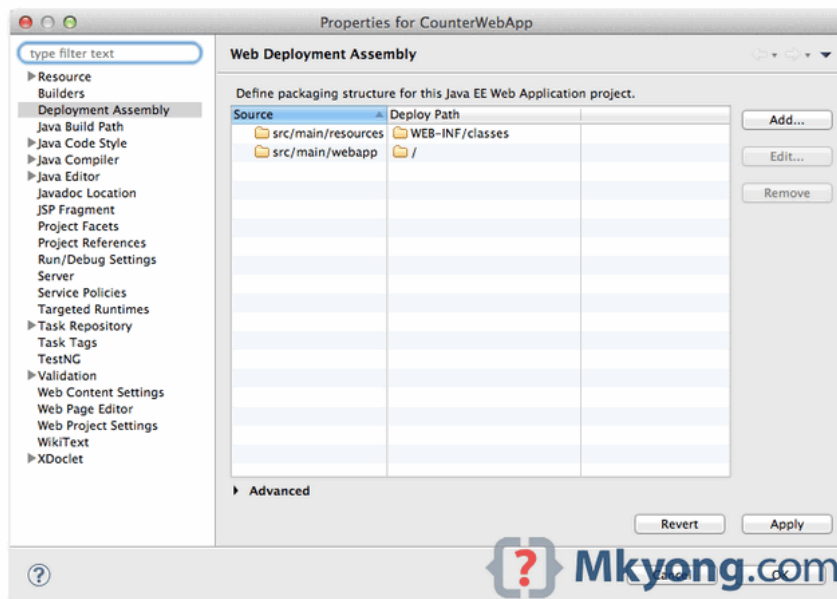


6. Eclipse + Tomcat

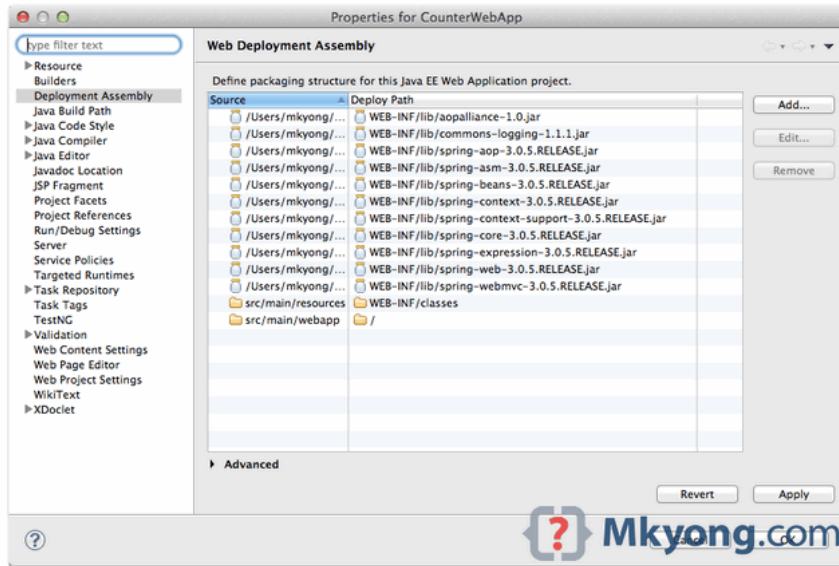
In order to start or debug this project via Eclipse server plugin (Tomcat or other container). You need to issue following command again, in order to make all dependencies attached to the project web deployment assembly.

```
mvn eclipse:eclipse -Dwtpversion=2.0
```

Before this command, project dependencies are empty.



After this command, now project dependencies are here!



Important!

Many developers are trapped here, and failed to perform the starting or debugging in Eclipse server plugin, all failed by showing dependencies not found error message. Right click on your project properties, make sure all dependencies are inside the web deployment assembly, otherwise issue `mvn eclipse:eclipse -Dwtversion=2.0` again!

7. Maven Packaging

Review the `pom.xml` again, the `packaging` tag defining what is the packaging format or output.

pom.xml

```
<project ...>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mkyong</groupId>
  <artifactId>CounterWebApp</artifactId>

  <packaging>war</packaging>
  <version>1.0-SNAPSHOT</version>
```

Package the project to deploy is easy, just issue `mvn package`, it compiles and package the web project into a "war" file and store in `project/target` folder.

For example :

```
$pwd
/Users/mkyong/Documents/workspace/CounterWebApp

Yongs-MacBook-Air:CounterWebApp mkyong$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building CounterWebApp Maven Webapp 1.0-SNAPSHOT
[INFO] -----
[INFO] -- omitted for readability
[INFO] --- maven-war-plugin:2.1.1:war (default-war) @ CounterWebApp ---
[INFO] Packaging webapp
[INFO] Assembling webapp [CounterWebApp] in [/Users/mkyong/Documents/workspace/CounterWebApp/target]
[INFO] Processing war project
[INFO] Copying webapp resources [/Users/mkyong/Documents/workspace/CounterWebApp/src/main/webapp]
[INFO] Webapp assembled in [87 msecs]
[INFO] Building war: /Users/mkyong/Documents/workspace/CounterWebApp/target/CounterWebApp.war
[INFO] WEB-INF/web.xml already added, skipping
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.936s
[INFO] Finished at: Thu Dec 20 22:28:53 MYT 2012
[INFO] Final Memory: 14M/206M
```

[INFO] -----

Done, just copy the `project/target/CounterWebApp.war` file and deploy to your container.

8. Demo

Start the web application.

`http://localhost:8080/CounterWebApp/welcome`



`http://localhost:8080/CounterWebApp/welcome/mkyong`



Download Source Code

Download it – [CounterWebApp.zip](#) (13 KB)

References

1. [Apache Maven Project](#)
2. [How To Create A Java Project With Maven](#)
3. [Spring 3 REST Hello World Example](#)
4. [A Simple Web Application](#)

Tags : [maven](#) [maven-project](#) [spring mvc](#) [web project](#)



mkyong

Founder of Mkyong.com, love Java and open source stuffs. Follow him on [Twitter](#), or befriend him on [Facebook](#) or [Google Plus](#).

Here are some of my favorite books



[Head First
Java](#)



[Design
Pattern](#)



[Effective
Java](#)



[Spring
Recipes](#)



[JavaScript
Ninja](#)

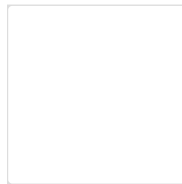
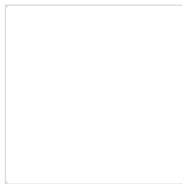
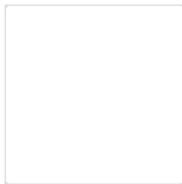
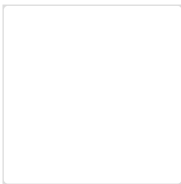
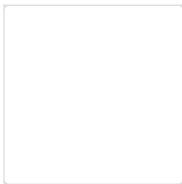
Related Posts

- ▶ [According to TLD, tag form:input must be empty, but is not](#)
- ▶ [Maven, JAVA_HOME is not defined correctly on Mac OSX](#)
- ▶ [Spring REST @RequestMapping extract incorrectly if value contains '.'](#)
- ▶ [How to create a Java project with Maven](#)
- ▶ [How to create a manifest file with Maven](#)

Popular Posts

- ▶ [Top 8 Java People You Should Know](#)
- ▶ [Top 20 Java Websites](#)
- ▶ [Top 5 Free Java eBooks](#)
- ▶ [Top 10 Java Regular Expression Examples](#)
- ▶ [Top 5 Open Source Q&A Systems](#)

You might also like following tutorials :



Piotrek

August 14, 2013 at 2:28 am

How to add directory for unit tests? I would like to have junit directory like following: src/test/java. I added it manually and included src/main/test in my pom file. Then I execute 'mvn clean install' command in project directory but no tests are run... Is there archetype to create web maven project with test directory?

[Reply](#)



phenomenon mcm

August 7, 2013 at 1:34 pm

?????!

[Reply](#)

**Rohan**

July 28, 2013 at 9:16 am

I want to know how you got the project structure, because whenever i do this command I get only src>main>resources and src>main>webapp. I don't get src>main>java. Do you have to do it by hand or maven generates for you.

[Reply](#)**Kittu**

July 27, 2013 at 9:15 pm

Hi mkyong,

I have followed the way to create the Eclipse web project, and I got the Eclipse web project ready to import in Eclipse, but I wanna know that

Am I suppose to import this project as General project or Maven project?
if I have to import it as maven project then tell me how to get "add maven project option" in eclipse, it is not coming in my case.

Note:- I am using Eclipse Galileo.

looking forward for your reply.
Thanks

kittu

[Reply](#)**Yan**

July 9, 2013 at 3:15 pm

Hi MK,

I am new to the tech. I have a problem for recognise the path of the spring source for my eclipse. It always say for example the C:/THE/PATH/TO/spring-aop-3.2.3.RELEASE.jar cannot be read or is not valid ZIP file. But I have see the file is existing in the folder.

Thanks
Yan

[Reply](#)**Martin Daum**

July 3, 2013 at 9:53 pm

Thanks MK for all these tutorials, they've been a great help on my path to JAVA programming proficiency (which I still haven't quite reached). I will recommend you and your site anytime I talk to a beginner JAVA developer seeking a jumpstart!

[Reply](#)**Ravikant Bhargava**

July 3, 2013 at 1:51 pm

Hi,

Though I have read your blogs/examples many times before but it is the first time I am posting here to 'Thank You' for saving me so much of time. Please accept my apologies for not doing so earlier. Infact it is not just the time saving part but the clarity of steps also helped me grasp some concepts which otherwise could have taken a long time to understand.

Taking a mental note to go through your other notes as well.
Keep up the good work. :)

Regards,
Ravikant Bhargava

[Reply](#)**Zwirro**

July 1, 2013 at 2:02 am

You are genius! I was fighting with eclipse, maven and spring for 10 hours and then found this page. 10 minutes and its working :D

[Reply](#)**Arif**

June 22, 2013 at 11:23 am

Hey MK,

Great tutorials!

when i came to this one, as soon as i do step-1, creating the mvn project, i get this error:

"The desired archetype does not exist (org.apache.maven.archetypes:maven-archetype-webapp:1.0)"

did anybody else got this? i am trying to find what's the issue here, but surely some help would be great!

[Reply](#)**piscoon**

June 20, 2013 at 9:57 pm

Hi mkyoung! Very good post. In "BaseController.java" you forgot to add "import org.springframework.web.bind.annotation.PathVariable;"

[Reply](#)**Abbin**

June 12, 2013 at 7:47 pm

Hi MK,

This is an excelent one. Thanks for that.

I just wanted to ask one query.

I could find a jetty-maven-plugin in you POM which is not mentioned anywhere here.

Could you explain that too. That would be helpful.

[Reply](#)**Abbin**

June 12, 2013 at 7:56 pm

to add: my jsp page was not loading before that. Worked fine after the addition

[Reply](#)**bruno melhor**

May 30, 2013 at 9:17 am

Hi Mk, great tutorial :D

but i faced a problem that i can solve.

I run the command generate-sources eclipse:eclipse, and after that when i refresh the project eclipse give the error:

/teste12/gen already exists but is not a source folder. Convert to a source folder or rename it.

I have a folder named WSDL with the devicemgmt.wsdl and onvif.xsd

The mvn -version:
Apache Maven 3.0.5 (r01de14724cdef164cd33c7c8c2fe155faf9602da; 2013-02-19 13:51:28+0000)
Maven home: C:\apachemaven3
Java version: 1.6.0_45, vendor: Sun Microsystems Inc.
Java home: C:\Program Files\Java\jdk1.6.0_45\jre
Default locale: pt_PT, platform encoding: Cp1252
OS name: "windows 7", version: "6.1", arch: "x86", family: "windows"

My pom file:

```
4.0.0
fr.norsys.asoape.it
teste12
1.0-SNAPSHOT
jar

1.1

fr.norsys.asoape
runtime-library
${asoape.version}

com.google.android
android
1.6_r2
provided

${project.artifactId}
src

fr.norsys.asoape
asoape-maven-plugin
${asoape.version}

generate-soap-stub

generate-sources

C:\Users\pc\workspace2\teste12\wsdl

com.jayway.maven.plugins.android.generation2
android-maven-plugin
3.0.0
true

target

${env.ANDROID_HOME}
17

true
true

alignApk
package

zipalign

org.apache.maven.plugins
maven-compiler-plugin
2.3.2

1.6
1.6

org.apache.maven.plugins
maven-eclipse-plugin
2.8

com.android.ide.eclipse.adt.ResourceManagerBuilder
com.android.ide.eclipse.adt.PreCompilerBuilder
com.android.ide.eclipse.adt.ApkBuilder
```

```
com.android.ide.eclipse.adt.AndroidNature

com.android.ide.eclipse.adt.ANDROID_FRAMEWORK
org.eclipse.jdt.launching.JRE_CONTAINER

true

com.google.android:android
commons-logging:commons-logging
org.apache.httpcomponents:httpclient
org.apache.httpcomponents:httpcore
commons-codec:commons-codec
org.khronos:opengl-api
xerces:xmlParserAPIs
xpp3:xpp3
```

Can you help me???

Thanks in advance

Reply



Sara

May 18, 2013 at 8:13 am

Everything was working until the last part of the tutorial. I tried to access

<http://localhost:8080/CounterWebApp/welcome/mkyong>

Then i got :

HTTP Status 500 – Request processing failed; nested exception is

org.springframework.web.bind.annotation.support.HandlerMethodInvocationException: Failed to invoke handler method [public java.lang.String

com.mkyong.controller.BaseController.welcomeName(java.lang.String,org.springframework.ui.ModelMap)]; nested exception is java.lang.IllegalStateException: No parameter name specified for argument of type [java.lang.String], and no parameter name information found in class file either.

Reply



Sara

May 21, 2013 at 3:51 am

Strange, i was able to access <http://localhost:8080/CounterWebApp/welcome/mkyong> after rebooting my laptop the next day.

Reply



Anubhab

May 14, 2013 at 3:01 pm

Hi Can you tell me how can you access the war version name in index.jsp??

Suppose the war i create is like appName-1.1.2.war.

I want to print the version i.e. "1.1.2" in my index.jsp.

How is that possible?

Reply



Dino

May 13, 2013 at 12:38 pm

The tutorial is awesome, it saves me lots of time on troubleshooting. Thank you so much.

BTW, I encountered error of "Java compiler level does not match the version of the installed Java project facet" when I tried the step 1 for the first time. I was still able to run the project, but had a red cross on the project. The solution can be found here, <http://stackoverflow.com/questions/7715260/java-compiler-level-does-not-match-the-version-of-the-installed-java-project-fac>

I am using Maven 3.0.5, jdk 1.7.0_10, Eclipse Java EE Juno. But later I was not able to reproduce it.

[Reply](#)

Surajit Bhattacharjee

May 8, 2013 at 9:45 am

Super useful article! Saved me hours, very nicely done MK!

[Reply](#)

Alket

May 2, 2013 at 7:01 pm

Hi, are you using Java EE for this example or just Java SE ?

[Reply](#)

[Read the Full Document](#)

April 18, 2013 at 12:59 pm

I have learn several just right stuff here. Definitely worth bookmarking for revisiting.
I surprise how a lot effort you put to make such a excellent informative web site.

[Reply](#)

chinmay

March 28, 2013 at 6:33 am

Hi ,
Can you let me know how to import a web project in eclipse ?I am not able to do it.

Regards
Chinmay

[Reply](#)

Cyril

April 1, 2013 at 5:17 pm

Hello,

if you want to import other project into eclipse just Right click on your project explorer and select import →
General—>Existing Projects into workspace, in this window you need to browse for the actual path where
your project has to imported

[Reply](#)

Pierluigi

March 20, 2013 at 8:06 pm

you are a cool guy MK, your tutorial really helped me getting started with Maven... all this XML is a nightmare anyway.
I am looking for alternatives because all that XML visual noise drives me crazy.

[Reply](#)

Maulana

March 18, 2013 at 5:50 pm

Hello,
I'm a newbie for eclipse and java. my background was Delphi and VB. now I'm learning java and eclipse. it is quite
interesting to build web application using maven. but what if I want to create java project using maven, what archetype

should I use, while in eclipse when I choose the maven project a lot of maven archetype can we select, and what if I separate the front end project (web application) and back end project (i.e for connecting to database) what should I do then.

thank you

Reply



Sarath

March 15, 2013 at 1:35 am

when i am trying to run the mvn package from eclipse i am getting the error as mentioned below, but when i am running from command prompt, its working fine can you please suggest how we can run through eclipse.

Error getting default plugin information: for project org.apache.maven.project.ProjectBuildingException: Error getting default plugin information: for project

```
at org.apache.maven.project.DefaultProjectBuilder.build(DefaultProjectBuilder.java:141)
at org.apache.maven.project.DefaultProjectBuilder.build(DefaultProjectBuilder.java:81)
at org.apache.maven.DefaultMaven.collectProjects(DefaultMaven.java:293)
at org.apache.maven.DefaultMaven.getProjectsForMavenReactor(DefaultMaven.java:241)
at org.apache.maven.DefaultMaven.execute(DefaultMaven.java:109)
at org.apache.maven.embedder.MavenEmbedder.execute(MavenEmbedder.java:534)
at org.apache.maven.cli.MavenCli.doMain(MavenCli.java:167)
at org.apache.maven.cli.MavenCli.main(MavenCli.java:61)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:37)
at java.lang.reflect.Method.invoke(Method.java:599)
at org.codehaus.plexus.classworlds.launcher.Launcher.launchEnhanced(Launcher.java:289)
at org.codehaus.plexus.classworlds.launcher.Launcher.launch(Launcher.java:229)
at org.codehaus.plexus.classworlds.launcher.Launcher.mainWithExitCode(Launcher.java:408)
at org.codehaus.plexus.classworlds.launcher.Launcher.main(Launcher.java:351)
at org.codehaus.classworlds.Launcher.main(Launcher.java:31)
Caused by: org.apache.maven.lifecycle.LifecycleExecutionException: Error getting default plugin information:
at
org.apache.maven.lifecycle.DefaultLifecycleExecutor.getDefaultPluginConfiguration(DefaultLifecycleExecutor.java:914)
at
org.apache.maven.lifecycle.DefaultLifecycleExecutor.populateDefaultConfigurationForPlugin(DefaultLifecycleExecutor.java:844)
at
org.apache.maven.lifecycle.DefaultLifecycleExecutor.populateDefaultConfigurationForPlugins(DefaultLifecycleExecutor.java:784)
at org.apache.maven.project.DefaultProjectBuilder.build(DefaultProjectBuilder.java:136)
... 16 more
Caused by: org.apache.maven.plugin.PluginResolutionException: Plugin could not be resolved: Missing:
-----
1) org.apache.maven.plugins:maven-jar-plugin:maven-plugin:2.2
```

Try downloading the file manually from the project website.

Then, install it using the command:

```
mvn install:install-file -DgroupId=org.apache.maven.plugins -DartifactId=maven-jar-plugin -Dversion=2.2 -
Dpackaging=maven-plugin -Dfile=/path/to/file
```

Alternatively, if you host your own repository you can deploy the file there:

```
mvn deploy:deploy-file -DgroupId=org.apache.maven.plugins -DartifactId=maven-jar-plugin -Dversion=2.2 -
Dpackaging=maven-plugin -Dfile=/path/to/file -Durl=[url] -DrepositoryId=[id]
```

1 required artifact is missing.

for artifact:

```
org.apache.maven.plugins:maven-jar-plugin:maven-plugin:2.2
```

from the specified remote repositories:

```
central (http://repo1.maven.org/maven2, releases=true, snapshots=false)
```

```
org.apache.maven.plugins:maven-jar-plugin:maven-plugin:2.2
```

from the specified remote repositories:

```
central (http://repo1.maven.org/maven2, releases=true, snapshots=false)
```

```
at org.apache.maven.plugin.DefaultPluginManager.loadPlugin(DefaultPluginManager.java:145)
at org.apache.maven.plugin.DefaultPluginManager.getMojoDescriptor(DefaultPluginManager.java:672)
```

at
org.apache.maven.lifecycle.DefaultLifecycleExecutor.getDefaultPluginConfiguration(DefaultLifecycleExecutor.java:906
... 19 more

[Reply](#)**Hilda**

March 7, 2013 at 11:51 pm

Hi, I do believe this is an excellent web site. I stumbled upon it ;)
I am going to revisit yet again since I bookmarked it.
Money and freedom is the greatest way to change, may you be rich and continue to guide other people.

[Reply](#)**[webhosting wordpress](#)**

March 4, 2013 at 8:19 am

I know this is off topic but I'm looking into starting my own weblog and was wondering what all is needed to get setup?
I'm assuming having a blog
like yours would cost a pretty penny? I'm not very internet smart so I'm not 100% positive. Any recommendations or
advice would be greatly appreciated. Thanks

[Reply](#)**Anonymous**

March 3, 2013 at 5:55 am

I'm really enjoying the theme/design of your site. Do you ever run into any internet browser compatibility issues? A
couple of my blog audience have complained about my blog not operating correctly in Explorer but looks great in
Chrome. Do you have any ideas to help fix this problem?

[Reply](#)**Hayk**

February 11, 2013 at 5:12 pm

Hi there is a little mistake in your project:
please add this line in your dispatcher.xml file

```
<context:annotation-config/>
```

otherwise controller is not being found...

[Reply](#)**Hayk**

February 11, 2013 at 4:48 pm

Hi, when I download your source zip folder and try to deploy it displays an error in console"

```
AM org.springframework.web.servlet.DispatcherServlet noHandlerFound  
WARNING: No mapping found for HTTP request with URI [/CounterWebApp/welcome] in DispatcherServlet
```

I suppose in mvc-dispatcher-servlet.xml there should be this line:

```
<mvc:annotation-driven/>
```

can you please help me?

[Reply](#)

**bobby**

April 3, 2013 at 8:55 am

i'm having the same issue. how do i fix this?? thanks!

[Reply](#)**raceconditions**

May 31, 2013 at 1:46 am

See Igor's problem below. You need to make sure your controller is in src/main/java and not in src/main/resources.

[Reply](#)**dli7mar**

January 31, 2013 at 6:17 am

I have a maven web project and defined tomcat7 as web server in eclipse. But I couldn't add this project to tomcat server because it is not in the available list.

How could I add a maven web project to eclipse tomcat server?

Many Thanks.

[Reply](#)**Amarjit**

January 29, 2013 at 6:21 pm

Hi MK for me .class and .project file is not created please help to resolve this problem asap

[Reply](#)**[Sijo](#)**

January 29, 2013 at 10:23 am

Hello sir, how can I create a Maven Java Application with Netbeans ?

[Reply](#)**Leo**

January 23, 2013 at 6:55 am

For convenience, we can declare maven-eclipse-plugin in file pom.xml to avoid typing the parameter "-Dwtpversion=2.0" when use command `mvn eclipse:eclipse`.

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-eclipse-plugin</artifactId>
  <version>2.8</version>
  <configuration>
    <downloadSources>true</downloadSources>
    <downloadJavadocs>false</downloadJavadocs>
    <wtpversion>2.0</wtpversion>
  </configuration>
</plugin>
```

[Reply](#)**[mkyong](#)**



January 23, 2013 at 9:21 am

Thanks for your invaluable input :)

[Reply](#)**Dave Potts**

January 12, 2013 at 7:51 pm

I tried this, when I tried to import it into Eclipse, eclipse bailed out with an internal error.

I then attempted to repeat the procedure from within eclipse, following the instructions from <https://docs.sonatype.org/display/M2ECLIPSE/WTP+mini+howto>, which is the official support site for the maven plugin used in this demo.

[Also includes instructions on how to create a java source folder]

Result Eclipse did not die and I can correctly configured eclipse/maven web project.

[Reply](#)**Igor**

December 31, 2012 at 3:41 pm

Hi MK,

maven-archetype-webapp creates directory structure without standard src/main/java folder. In this tutorial i can't find part where src/main/resources is renamed to src/main/java.

Thanks,
Igor

[Reply](#)**[mkyong](#)**

January 1, 2013 at 8:27 am

Create the folder structure manually :)

[Reply](#)**kasi**

May 3, 2013 at 10:56 am

this should be done, else the target has the java file not the class files. In 'basecontroller' also import `import org.springframework.web.bind.annotation.PathVariable;`
for a new bee every thing should be same else we need to spend hours in making the project work.

thanks for your work, mkyong

[Reply](#)**[yasser](#)**

December 21, 2012 at 8:17 pm

thank you very much very nice and useful lesson .i bookmark this site ,i love it coz i want learn programming .. thanks alot

[Reply](#)**dli7mar**

November 18, 2012 at 11:31 pm

Hi mkyong,

I have a maven3/jpa2/tomcat7/eclipse project, which has these jpa related dependencies in its pom.xml:

```
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-entitymanager</artifactId>
    <version>3.6.5.Final</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>org.hibernate.javax.persistence</groupId>
    <artifactId>hibernate-jpa-2.0-api</artifactId>
    <version>1.0.0.Final</version>
</dependency>
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>3.6.7.Final</version>
</dependency>
```

After running 'mvn eclipse:eclipse' command, I could see hiberante-entitymanager-3.6.5.Final.jar in the project. But after running 'mvn package' commeand, this jar file did not go \$my_project\target\my_project\WEB-INF\lib fold. When I deply the war file to tomcat, I need to manually copy/paste this jar file to tomcat to run it.

Do you know why this issue happened? Any idea is greatly appreciated.
Thank you so much.

Reply



mkyong

November 19, 2012 at 3:04 pm

Delete the "scope" tag, "provided" means your container will provide this jar, so it will not package into the war file.

```
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-entitymanager</artifactId>
    <version>3.6.5.Final</version>
</dependency>
```

Reply



dli7mar

November 19, 2012 at 9:52 pm

Your way works perfect!
Thank you so much!

Reply



jssay

November 4, 2012 at 3:30 pm

thanks for your posting, this is very detailed and helpful. thanks again.

Reply



Racs

September 8, 2012 at 1:37 pm

Hi Mkyong, this is cool! Thanks for a very detailed (and working) solution. :-)

Reply

**dli7mar**

August 25, 2012 at 4:23 am

Hi, will you please explain the details of running in tomcat?
>In Eclipse, create a server instance, add above "primefaces" web project and start it.

[Reply](#)**pradeep**

June 23, 2012 at 11:22 am

i did same what you have written in the above but the directory mkyongweb-core is created but in that folder webapp->WEB-INF-> web.xml is not created.

why is not created ?
How can i resolve this problem?

Thanks,
Pradeep

[Reply](#)**[mkyong](#)**

August 3, 2012 at 1:31 pm

Just tested the command again, and the web.xml is created successful. May be you can try the command again, or send me your command.

[Reply](#)**pigx**

August 10, 2012 at 7:17 pm

The command is right. But you need to remove the invisible NL char between the second and first line.

[Reply](#)**Gagandeep Singh**

April 10, 2012 at 9:41 pm

using the -DgroupId=com.store.controllers here.. was needless since it didn't create the package although the webApp was successfully created.

```
mvn archetype:generate -DgroupId=com.store.controllers -DartifactId=shoppersStop -DarchetypeGroupId=org.apache.maven.plugins -DarchetypeArtifactId=maven-archetype-webapp
```

I would say .. that the project folder structure was partially created ..
well but at least I am happy that I am learning maven no matter if in parts.. haha..

But thanks .. mkyong!!! you are the best when it comes to practising technologies.

[Reply](#)**[mkyong](#)**

August 3, 2012 at 1:32 pm

Agreed with you, Maven should generate the packaging folder as well.

[Reply](#)**Rahul**

August 6, 2012 at 7:50 pm

Hi Mkyong

can you please tell me how can i generate package structure with web project

Reply



Radha

April 1, 2012 at 10:47 pm

Hey MK you are awesome. Thanks for these tutorials they have saved me so much time!

Reply

What is your opinion?

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

Note

To post source code in comment, uses `<pre lang="java"> source code </pre>`

Post Comment

☐ Notify me of followup comments via e-mail

Create a Presentation Now

 [google.com/drive](https://drive.google.com)

Create Free Presentations Online With Google Slides. Try Now - Free!



AdChoices 

All Available Tutorials

Java Core Technologies :

Java I/O, Java RegEx, Java XML, Java JSON, JDBC, Java Misc

J2EE Frameworks :

Hibernate, JSF 2.0, Spring Core, Spring MVC, Spring Security, Spring MongoDB, Spring BatchApache Wicket, Struts 1.x, Struts 2.x

Web Service :

JAX-WS (SOAP), JAX-RS (REST)

Build Tools :

Maven, Archiva

Unit Test Frameworks :

jUnit, TestNG

Others :

Android, Google App Engine, jQuery, Java MongoDB, Quartz Scheduler

Favorites Links

Android Developer

Google App Engine using Java

DZone - Fresh Links

Official Java EE 5 Tutorial

Official Java EE 6 Tutorial

Spring 2.5.x documentation

Spring 3.2.x documentation

Hibernate core documentation

Java SE 6.0 API documentation

Java EE 6.0 API documentation

Java Secure Socket Extension

(JSSE) Reference Guide

JSP home page

JSF home page

Eclipse IDE for Java developer

Struts 1.3 documentation

Struts 2.2 documentation

Friends & Links

Java Code Geeks

PHP Tutorials

TenthOfMarch

Web Security Blog

Web Development

About Us

Mkyong.com is a weblog dedicated to Java/J2EE developers and Web Developers. We constantly publish useful tricks, tutorials on J2EE or web development.

All examples are simple, easy to read, and full source code available, and of course well tested our development environment.

We're Social

Twitter - [Follow Me](#)

Facebook - [Like Me](#)

Google Plus - [Add Me](#)

RSS - [Subscribe Me](#)

[Maven home page](#)

[Maven central repository Search](#)

[Java.Net Maven repository](#)

[Ant home page](#)

[JAX-WS Official Website](#)

[JAX-RS Official Website \(Jersey\)](#)

[MongoDB Official Website](#)

Copyright © 2008-2013 Mkyong.com, all rights reserved. Web Hosting Powered by Liquid Web