

# Spring MVC HelloWorld Using Maven in Eclipse

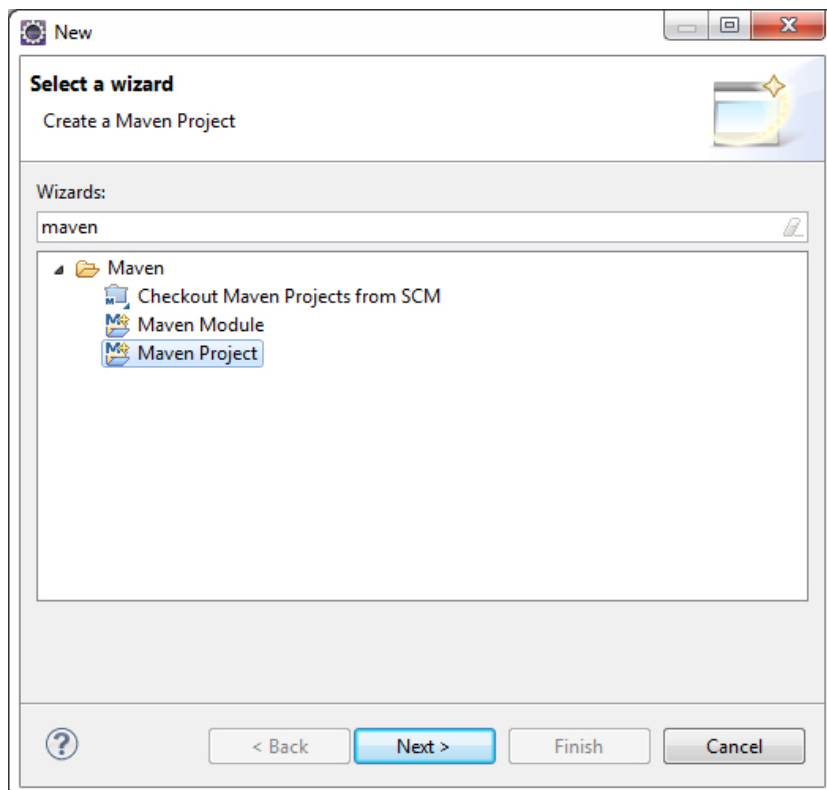
Java developers often rely on examples to learn Spring framework. Simple examples are often a key learning resource. There are many Spring MVC HelloWorld applications. However, most of them are outdated (do not integrate Maven, use old version of Spring, etc) or not complete (missing key steps or file hierarchy view). Therefore can not lead to a perfectly working Hello World. In this article, I will show steps for creating a Spring MVC HelloWorld application using Maven in Eclipse. All files's content and locations are illustrated.

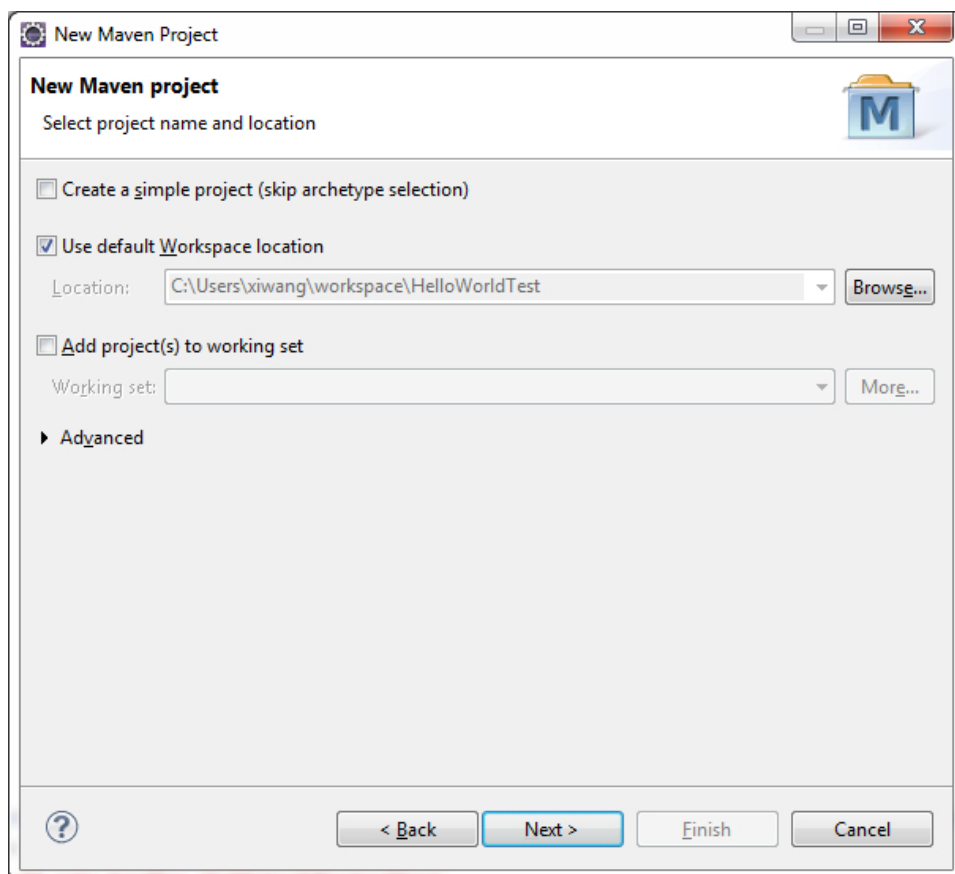
## Prerequisite

- » Windows 7, Java 7, Eclipse Kepler Java EE
- » Tomcat 7 ([How to install Tomcat 7 on Eclipse](#))

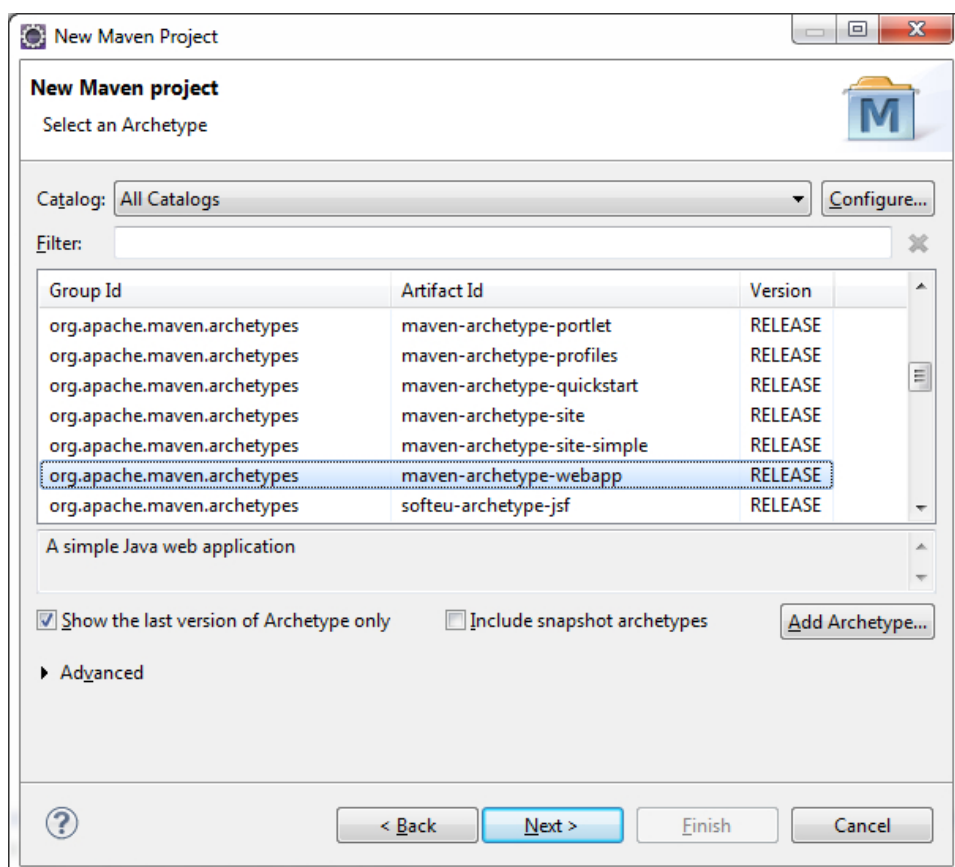
## Step 1: Create a Maven Project

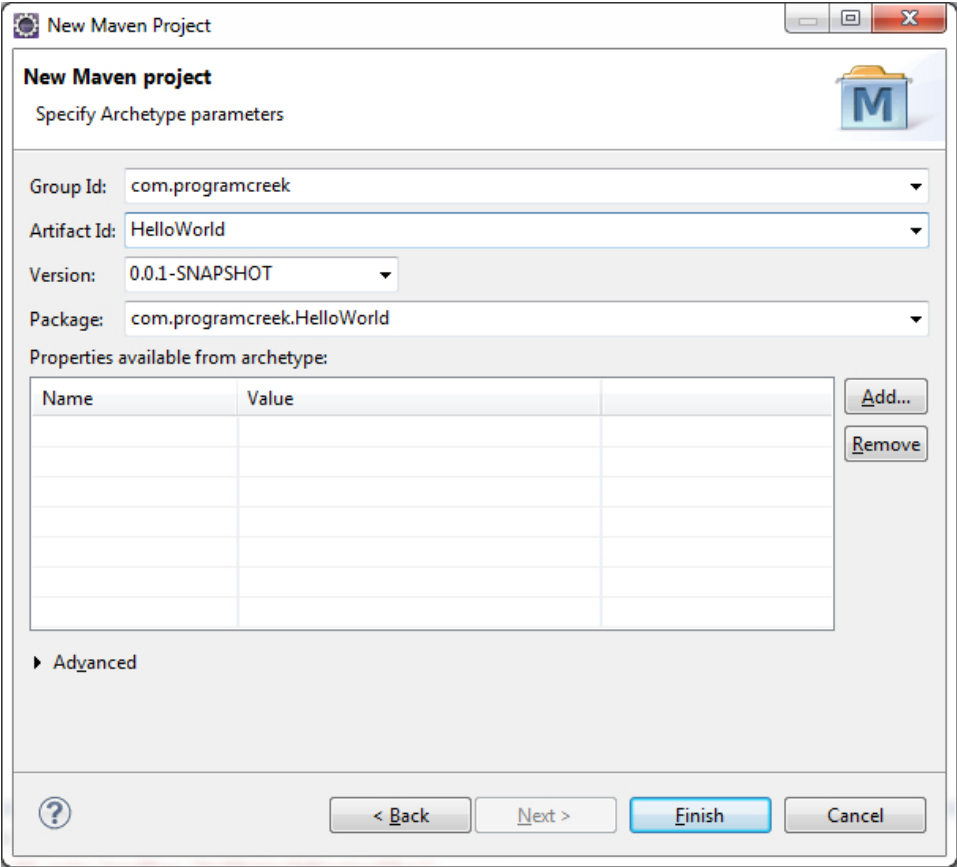
Create a Maven project by following the following steps:





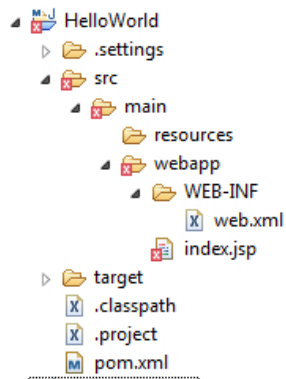
Select "webapp".



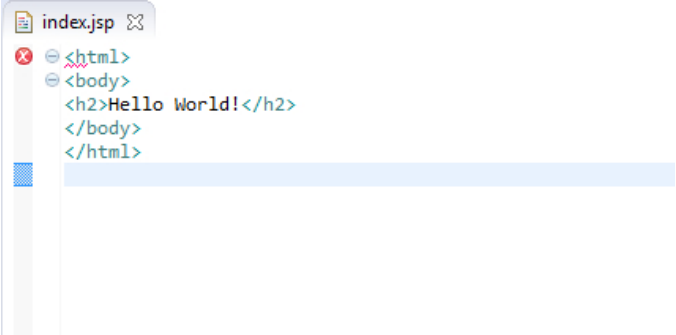


GroupId identifies the project uniquely across all projects, so we need to enforce a naming schema. ArtifactId is the name of the jar without version. For more information about each field, check out the [official page](#) about maven naming convention. (You can do this later. It makes more sense, when you complete the project first.)

After the Maven project is created, the project in the Navigator view should look like the following:



As shown above, there is an error marked with red. If you open index.jsp file, you can see the error message:



The superclass "javax.servlet.http.HttpServlet" was not found on the Java Build Path

To fix the problem, right click on project -> Properties -> Java Build Path -> Add Library...-> Server Runtime -> Apache Tomcat -> Finish.

## Step 2: Configure Spring

To make a Spring web application, we need to configure several xml files. First of all, we need to add Spring dependencies. Edit the automatically generated pom.xml file to be the following:

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.programcreek</groupId>
    <artifactId>HelloWorld</artifactId>
    <packaging>war</packaging>
    <version>0.0.1-SNAPSHOT</version>
    <name>HelloWorld Maven Webapp</name>
    <url>http://maven.apache.org</url>

    <properties>
        <spring.version>4.0.1.RELEASE</spring.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
        <!-- Spring 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>

    </dependencies>

    <build>
        <finalName>HelloWorld</finalName>
    </build>
</project>
```

Edit the default web.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
```

```

http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">

<display-name>Archetype Created Web Application</display-name>

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

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

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

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

```

Create an xml file "dispatcher-servlet.xml" under the same directory of web.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.programcreek.helloworld.controller" />

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

```

In the above xml file, base-package specifies the package of the controllers. prefix specifies the directory of views, and it is set to be /WEB-INF/views/, which means views directory should be created under WEB-INF. suffix specifies the file extension of views. For example, given a view "helloworld", the view will be located as /WEB-INF/views/helloworld.jsp. You can figure this out later in Step 3.

## Step 3: Create Spring Controller and View

Create the HelloWorldController under src/main/java/ directory.

HelloWorldController.java

```

package com.programcreek.helloworld.controller;

```

```

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class HelloWorldController {
    String message = "Welcome to Spring MVC!";

    @RequestMapping("/hello")
    public ModelAndView showMessage(
        @RequestParam(value = "name", required = false, defaultValue = "World")
        String name) {
        System.out.println("in controller");

        ModelAndView mv = new ModelAndView("helloworld");
        mv.addObject("message", message);
        mv.addObject("name", name);
        return mv;
    }
}

```

In the code above, `@RequestMapping` annotation maps web requests onto specific handler classes and/or handler methods, in this case, `showMessage()`. It provides a consistent style between Servlet and Portlet environments, with the semantics adapting to the concrete environment.

`RequestParam` indicates that a method parameter should be bound to a web request parameter. In this case, we also make it not required and give it a default value.

`new ModelAndView("helloworld")` determines that `helloworld` is the target view.

`index.jsp`

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Spring 4 MVC - HelloWorld Index Page</title>
</head>
<body>

    <center>
        <h2>Hello World</h2>
        <h3>
            <a href="hello?name=Eric">Click Here</a>
        </h3>
    </center>
</body>
</html>

```

Create the `helloworld.jsp` file under `/WEB-INF/views/` directory.

`helloworld.jsp`

```

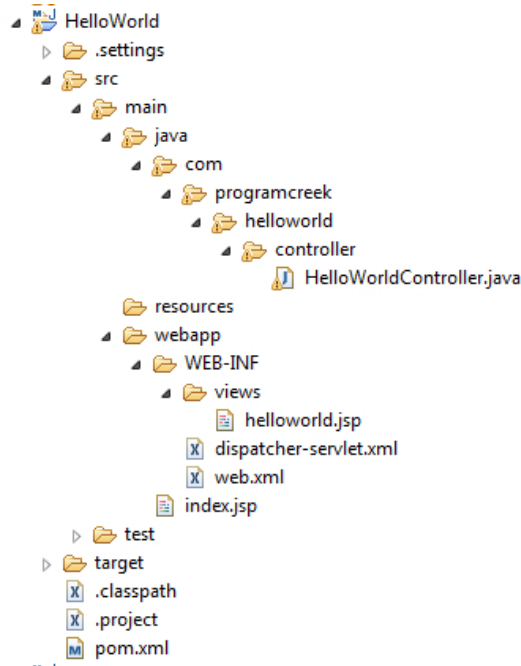
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

```

```
<title>Spring 4 MVC -HelloWorld</title>
</head>
<body>
    <center>
        <h2>Hello World</h2>
        <h2>
            ${message} ${name}
        </h2>
    </center>
</body>
</html>
```

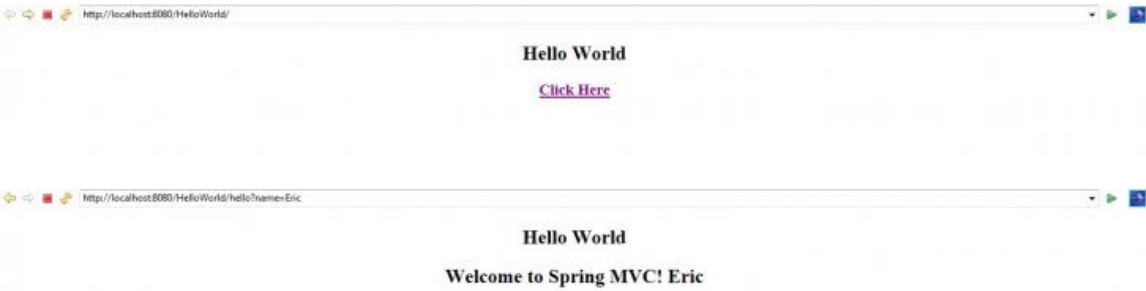
`${variable}` will be converted to the value of the variable.

Final Navigator view of the project:



## Step 4: Run on Server

Right click the project and select Run As --> Run on Server.



[Project zip file for download.](#)

[Next: Dependency Injection and XML Configuration](#)

### Related Posts:

- 1. [Spring MVC Tutorial – Setter Dependency Injection](#)
- 2. [Spring Framework Tutorial – Hello World](#)

- 3. Spring HelloWorld Example Using Eclipse and Maven
- 4. Spring MVC Tutorial: Process Form Submission

Category >> Spring

If you want someone to read your code, please put the code inside `<pre><code>` and `</code></pre>` tags. For example:

```
<pre><code>String foo = "bar";</code></pre>
```

- *Liam*  
(Sorry, some “important codes” lost after post, so try again)  
I’m beginner to SpringMVC and meet following two questions and resolved:  
□——(questions)——  
1. See the console “in controller” in Eclipse, but got HTTP Status 404 in my browser  
2. Still show \${message} and \${name} (jsp Expression Language network)  
□——(solutions)——  
1. Wrong folder name:  
In my “dispatcher-servlet.xml”

/WEB-INF/views/

but I named my folder under WEB-INF “view”,  
so just resolved by changing the “views” in tag to “view”.

- 2. Add to helloworld.jsp  
In the first line of “helloworld.jsp”, add  
You can see more information here:  
<http://www.codejava.net/java-ee/jsp/activating-and-deactivating-el-evaluation-in-jsp>  
□——  
Hope it helps

- *Liam*  
I’m beginner to SpringMVC and meet following two questions and resolved:  
□——(questions)——  
1. HTTP Status 404  
2. still show \${message} and \${name} (jsp Expression Language network)  
□——(solutions)——  
1. Wrong folder name:  
In my “dispatcher-servlet.xml”

/WEB-INF/views/

but I named my folder under WEB-INF “view”,



so just resolved by changing the “views” in tag to “view”.

2. Add to helloworld.jsp

In the first line of “helloworld.jsp”, add

You can see more information here:

<http://www.codejava.net/java-ee/jsp/activating-and-deactivating-el-evaluation-in-jsp>

□

Hope it helps

• *Eddy*

When the model values passed in the controller are not visible on the generated page, thus you get

`#{message} #{name}` instead of “Welcome to Spring MVC! Eric”,

it might be that your web.xml is not of the proper version. In my case it was version 2.3 (the web.xml was created automatically by eclipse) and had to upgrade it to version 3.

(note, I am using Spring MVC version 4.3.11 and Tomcat version 8.5.20).

My web.xml now begins with:

• *Kalyan Bhattacharya*

Excellent Article! Author explains the steps so well. Keep up the good work. I look forward to more such articles from you. It worked first time, I tried and taught me the basics.

Regards

Kalyan

• *Manuela*

Very useful tutorial for beginners.Thanks!

• *Poomchai Chokpipatporn*

Thank you for your tutorial , this is my first maven project with success

• *Rohit Sharma*

check pom.xml

•

<https://uploads.disquscdn.com/images/d5a57053f42dcee6e1120de23b49a6a32b8b8c812278b8200e699468b40f1f3a.png>

Can you help me, I got error

• *ccna training in pune*

Thanks for posting this useful content, Let me share this, . CCNA training in pune

• *ccna training in pune*

Thanks for posting this useful content, Good to know about new things here, Let me share this, . CCNA training in pune

• *Aayush Thakkar*

/WEB-INF/views/

.jsp

Change this dispatcher-servlet.xml to this

- *amitgb*

Thanks for the post. A quickstarter to understand basic examples :-).

Thanks again !!

- *amitgb*

Thank you very much. It worked well

- 贱贱的梦想

It's very useful,thanks.

- *toto*

thanks

- *omar*

-1:-1 31:3 The declaration for the entity “HTML.Version” must end with ‘>’. does anyone knows how to solve this ?

- *Srikanth*

Thank you very much. Very informative

- *jiemin*

nice tutorial!

- *Sujeeth Kumar*

Getting This error

Tomcat 8

quick response would be appreciated

Jan 27, 2016 4:48:47 PM org.apache.catalina.core.StandardWrapperValve invoke

SEVERE: Servlet.service() for servlet [dispatcher] in context with path [/SPRINGMVC1] threw exception [Servlet execution threw an exception] with root cause

java.lang.NoSuchMethodError: org.springframework.http.HttpMethod.matches(Ljava/lang/String;)Z

at org.springframework.web.servlet.FrameworkServlet.service(FrameworkServlet.java:841)

at javax.servlet.http.HttpServlet.service(HttpServlet.java:729)

at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:291)

at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)

at org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)

at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:239)

```
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:212)
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:106)
at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:502)
at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:141)
at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:79)
at org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.java:616)
at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:88)
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:521)
at org.apache.coyote.http11.AbstractHttp11Processor.process(AbstractHttp11Processor.java:1096)
at org.apache.coyote.AbstractProtocol$AbstractConnectionHandler.process(AbstractProtocol.java:674)
at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1500)
at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.run(NioEndpoint.java:1456)
at java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
at java.lang.Thread.run(Unknown Source)
```

- *STILLPHENOM* .

Getting this error.. any help is appreciated.

HTTP Status 404 – /HelloSpring/

type Status report

message /HelloSpring/

description The requested resource is not available.

Apache Tomcat/8.0.28

- *STILLPHENOM* .

i know this is a little old one but i am getting error on \${message} \${name}. could someone help out please.  
thanks

- *Shivam Singh*

well explain and must visit Spring MVC Example

- *Trongnhan Huynh*

when i build tomcat,the tomcat localhost log:Error configuring application listener of class  
org.springframework.web.context.ContextLoaderListener

- *Steve Nguyen*

\${variable} is not converted to the value of the variable in my case. Am I missing something? I followed all steps carefully.. Can you please help?

- *Kuki*

When I create the Maven project in Eclipse, the .classpath xml file and .project xml files are not generated. Also I this error. Which spring version did you write in pom.xml?

Pls help

- *Kuki*

When I create the Maven project in Eclipse, the .classpath xml file and .project xml files are not generated. Also I this error. Pls help.

- *aditi bhalerao*

I have performed all the steps as suggested but, I am getting page not found error when I click on the link which goes to helloworld.jsp. Actually, it gets navigated to the helloworld.jsp page, but, shows the following error on console:

Servlet /HelloWorldSpring threw load() exception

java.lang.NoSuchMethodError:

org.springframework.core.convert.converter.ConverterRegistry.addConverter(Ljava/lang/Class;Ljava/lang/Class;Lorg/springframe

Kindly help.

- *Apurva*

Awesome work dude...but u skipped first time server setup part which took a lot of time of mine...anyways great work !!!

- *Sheela Stephen*

Thanks.

When I am trying to work this initially I met some problems. But once made correct entries in pom.xml file, this example is working perfectly.

Important point. : you are missing to add spring context enteries in the pom.xml file. Please add that.

Final pom file I have used is,

```
4.0.0
com.madrone.lmsapp
lmsapp
war
0.0.1-SNAPSHOT
lmsapp Maven Webapp
http://maven.apache.org
```

junit

junit

3.8.1  
test

org.springframework  
spring-core  
4.1.4.RELEASE

org.springframework  
spring-web  
4.1.4.RELEASE

org.springframework  
spring-webmvc  
4.1.4.RELEASE

org.springframework  
spring-context  
4.1.4.RELEASE

lmsapp

- *Neha*  
Thanks a lot for the help..ya web.xml that josh mentioned is needed to rectify.

- *minsuk*  
I solved. My tomcat setting was wrong, I attach resources that I used.

<http://stackoverflow.com/questions/2280064/tomcat-started-in-eclipse-but-unable-to-connect-to-http-localhost8085>

<http://stackoverflow.com/questions/4919846/why-tomcat-server-location-property-is-greyed-in-eclipse>

- *minsuk*  
HTTP Status 404 – /HelloWorld/

type Status report

message /HelloWorld/

description The requested resource is not available.

Pivotal tc Runtime 3.0.2.SR2/8.0.15.B.RELEASE

Please help me to resolve this problem.

- *Anshul Zunke*  
Superb article to get started with Spring and Maven.

- *Aji Nurul Yakin*  
Thank You... Help full for me....

- *lianliang*

It's really useful to me thanks ...

- *Lian*  
I think you should configure server about tomcat....
- *Andy*  
Excellent tutorial moving from Netbeans to Eclipse. Very good starting point. Great work.
- *RakeshM*  
Hello Sebastian  
  
you might be using an old version of JSP 1.2 , please use JSP 2.0 ( check declaration in web.xml )  
  
or  
  
use following directive in jsp
- *Sebastian*  
Hi I run it on server and i get : `${message} ${name}` instead of "Welcome to Spring MVC! Eric" what could be the problem here ?
- *heena*  
HTTP Status 404 – /HelloWorld/  
  
type Status report  
  
message /HelloWorld/  
  
description The requested resource is not available.  
  
Pivotal tc Runtime 3.0.2.SR2/8.0.15.B.RELEASE  
  
Please help me to resolve this problem.
- *yuan*  
this post is still a nice tutorial for me, thanks
- *Sridhar Bandlamudi*  
src/main/java is not available in my project how to deal with this problem?
- *Kai Huang*  
Hi Chris,  
  
I guess you didn't set dispatcher-servlet.xml correctly. If you make a project with your own name, please modify the with your package to link the HelloWorldController.java.  
  
Best regards.
- *Chris*  
Hello, I get error  
No mapping found for HTTP request with URI [/WebDemo/hello] in DispatcherServlet with name 'dispatcher'

whats wrong?

- *Vineet Garg*  
pls check –  
<http://stackoverflow.com/questions/4166882/run-on-server-option-not-appearing-in-eclipse>
- *Sunil Joshi*  
Nice Tutorial. Really helpful.Thanks
- *Supriya Harish*  
When I right click on project run on server is not enabled. PLease help
- *Vu Nguyen*  
Thanks for a useful post.  
But in my case, I had to remove id="WebApp\_ID" from web.xml so that my Geronimo server run with name HelloWorld, not WebApp\_ID.
- *ceeSpotRun*  
Nice tutorial but you shouldn't have to worry about these types of things anymore... it's time consuming. Use the Jigy Generator to create your project with Spring completely configured for you. Jigy Generator reverse engineers your database to create all your DAO's, domain objects and validators... Plus you have certain things that just work out of the box like login, authentication, file upload etc. Jigy Generator can be downloaded at <http://www.getjigy.com>
- *Alex Malmann Becker*  
Thanks for your help.
- *Mario*  
Thanks a lot! This was really usefull!
- *Josh Lockhart*  
Please note that your tags are out of order in the web.xml file you specify, which causes Eclipse to complain.  
To get it to work, I had to re-order them according to the DTD  
([http://svn.apache.org/repos/asf/beehive/trunk/netui/external/struts/web-app\\_2\\_3.dtd](http://svn.apache.org/repos/asf/beehive/trunk/netui/external/struts/web-app_2_3.dtd)) to look like this:

Archetype Created Web Application

contextConfigLocation

/WEB-INF/dispatcher-servlet.xml

org.springframework.web.context.ContextLoaderListener

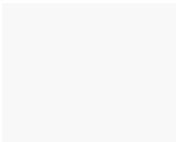
dispatcher

org.springframework.web.servlet.DispatcherServlet

1

dispatcher

/



Copyright © 2008 - 2018Program Creek