# Eclipse - java.lang.ClassNotFoundException: org.springframework.web.servlet.DispatcherServlet in Tomcat

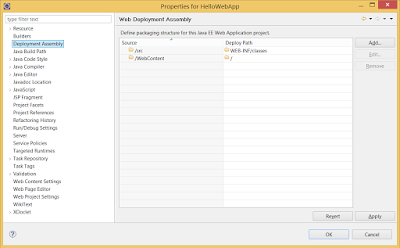
Problem: You are getting java.lang.ClassNotFoundException exception complaining that spring’s DispatcherServlet class is not available in class path. This error is coming while running a Spring MVC based Java application from Eclipse and Tomcat as Server (running inside Eclipse IDE itself). You have either included spring framework JAR files manually by yourself or you are using Maven to download and manage dependent JAR files. Here is the stack trace of this error :  
  
java.lang.ClassNotFoundException: org.springframework.web.servlet.DispatcherServlet  
    at org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1678)  
    at org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1523)  
    at org.apache.catalina.core.DefaultInstanceManager.loadClass(DefaultInstanceManager.java:525)  
    at org.apache.catalina.core.DefaultInstanceManager.loadClassMaybePrivileged(DefaultInstanceManager.java:507)  
    at org.apache.catalina.core.DefaultInstanceManager.newInstance(DefaultInstanceManager.java:126)  
    at org.apache.catalina.core.StandardWrapper.loadServlet(StandardWrapper.java:1099)  
    at org.apache.catalina.core.StandardWrapper.load(StandardWrapper.java:1043)  
    at org.apache.catalina.core.StandardContext.loadOnStartup(StandardContext.java:4957

Cause of java.lang.ClassNotFoundException: org.springframework.web.servlet.DispatcherServlet

There are two main cause of getting java.lang.ClassNotFoundException: org.springframework.web.servlet.DispatcherServlet error in Eclipse :  
  
1) first either you don't have **spring-webmvc-3.1.0.RELEASE.jar** or similar JAR, depending upon which Spring framework version you are using e.g. Spring 4.0  
  
2) Second is, Eclipse is not able to find this dependency in CLASSPATH.  
  
  
But before going into those, let's first understand  
**what is org.springframework.web.servlet.DispatcherServlet class?** and **why it is required?**.  
  
This class is actually the front controller in Spring MVC framework, all [HTTP requests](http://javarevisited.blogspot.sg/2012/03/get-post-method-in-http-and-https.html) goes via this class to different controller classes. You define this class to map against a URL pattern in web.xml file using [load-on-startup tag](http://javarevisited.blogspot.sg/2011/12/load-on-startup-servlet-webxml-example.html), that's why you see that **loadOnStartup()** call on stack-trace of this error.   
  
  
This is why when you deploy your Spring MVC based web application in Tomcat, the WebAppClassLoader search for org.springframework.web.servlet.DispatcherServlet class in class loader's CLASSPATH, which is WEB-INF/lib folder of your Java Web application.  
  
  
If it doesn't found this class inside any JAR file in WEB-INF/lib, it throws **java.lang.ClassNotFoundException: org.springframework.web.servlet.DispatcherServlet**.  
  
  
Now let's see how to solve this problem.

How to Fix java.lang.ClassNotFoundException: org.springframework.web.servlet.DispatcherServlet in Tomcat and Eclipse

Here are some steps you can take to deal with this error in Tomcat inside Eclipse IDE :  
  
1) If you are using Maven and if spring-webmvc is not added as dependency in your project's pom.xml then please add that. Once you do that, build your project using mvn install or mvn build. This will download spring MVC JAR files and other dependency.  
  
  
2) Second step is to add Maven Dependency into deployment assembly. This is required to move all dependent JAR files into WEB-INF/lib directory of Java web application.  
  
In order to add Maven Dependency into deployment assembly in Eclipse, follow below steps :  
  
i) Right click your project, select properties and choose Deployment Assembly.

[](http://3.bp.blogspot.com/-MbtCRqhMsDw/VfzcJA1JvUI/AAAAAAAADyE/5x0qUP1WS_E/s1600/How%2Bto%2Bopen%2Bdeployment%2Bassembly%2Bin%2BEclipse%2BJavaEE.png)

ii)  Now click Add and choose Java Build Path Entries and select Maven dependency their. If your project is not set for automatic build then just refresh the project and build it.  
  
Now you should be able to run your Spring MVC application in attached Tomcat from Eclipse IDE itself.  
  
3) If this doesn't fix your problem then do a Maven clean, followed by Maven build to regenerate WAR file.  
  
  
That's all about **how to fix java.lang.ClassNotFoundException: org.springframework.web.servlet.DispatcherServlet error in Eclipse and Tomcat**. It's also possible that you get this error outside Eclipse IDE, when you are deploying your WAR file in other environment. That time you must check whether relevant JAR file e.g. **spring-webmvc-3.1.0.RELEASE.jar** exists in either WEB-INF/lib directory of your application or Tomcat/lib directory. It's better to keep in WEB-INF/lib directory because when you undeploy all the classes loaded by **WebAppClassLoader** are eligible for garbage collection, which is not the case for JAR files loaded from Tomcat's lib directory. That can cause PermGen Memory leak in Tomcat.

Read more: <http://javarevisited.blogspot.com/2015/09/eclipse-javalangclassnotfoundexception.html#ixzz3ngjnivFJ>