# [DispatcherServlet and web.xml in Spring Boot](https://stackoverflow.com/questions/43225549/dispatcherservlet-and-web-xml-in-spring-boot)

<https://stackoverflow.com/questions/43225549/dispatcherservlet-and-web-xml-in-spring-boot>

I'm currently trying to move my project from Java EE to Spring Boot project. However, i've been stucked and confused on the part with dispatcher servlet and web.xml and it seems like web.xml is no longer being read by the project anymore. The current project is running on tomcat 7.

In my web.xml file, I have lots of servlet, servlet-mapping, filter and filter mapping and I don't really understand how to do the mapping in the dispatcher.

I've attached a sample of my web.xml below and the version is 2.5.

<?xml version="1.0" encoding="UTF-8"?>

<web-app metadata-complete="true" version="2.5" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd" 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">

<display-name>displayName</display-name>

<description>description</description>

<resource-ref>

...

</resource-ref>

<filter>

<filter-name>Some Filter Name</filter-name>

<filter-class>Some Filter Class</filter-class>

<init-param>

<param-name>Some Param Name</param-name>

<param-value>Some Value</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>Some Filter Name</filter-name>

<url-pattern>Some url-pattern</url-pattern>

</filter-mapping>

<context-param>

<param-name>Some Param Name</param-name>

<param-value>Some Param Value</param-value>

</context-param>

<servlet>

<servlet-name>Some Servlet Name</servlet-name>

<servlet-class>Some Servlet Class</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Some Servlet Name</servlet-name>

<url-pattern>Some Url Pattern</url-pattern>

</servlet-mapping>

</web-app>

 Run code snippet

Expand snippet

**Qns:**

1. Should I convert all the stuff in my web.xml to rely on the spring dispatcher, if yes how can I achieve that?
2. Is moving away from the web.xml the way to go for spring boot project?

Can anyone please guide me along here? Thanks!!

1. A: Yes, spring boot no longer relay on xml configuration and it configures an equivalent to the dispatcher servlet automatically. You can follow the following link to see how to register your filters: [How to add a filter class in Spring Boot?](https://stackoverflow.com/a/30658752/4473822)
2. If you use maven and not gradle, the only XML in your spring boot project should be pom.xml. The way to go with spring boot is moving all your xml configuration, web.xml etc to spring boot's auto configuration + your java configuration.

Spring boot works very good when you do everything in java configuration and follow its principals. From my experience with it, when you start merging XML configuration and the legacy spring it starts breaking the auto configuration process and its much better to try as much as you can to comply with the new spring boot best practices.

[WEB-INF folder is not creating in Spring Boot application?](https://stackoverflow.com/questions/48277952/web-inf-folder-is-not-creating-in-spring-boot-application)

<https://stackoverflow.com/questions/48277952/web-inf-folder-is-not-creating-in-spring-boot-application>

Q: I am creating a Spring Boot application, using Spring Initializr, But the folder structure is bit different, there is no WEB-INF folder, so where I will put the JSP files, shall I create the WEB-INF folder manually?

A: Usually the web files will go into the resources folder with spring boot. <https://spring.io/guides/gs/serving-web-content/>

Dumping the jsp files in src/main/webapp folder worked for me, both with or without WEB-INF. I followed the exact project structure mentioned in

1. [mkyong.com](http://www.mkyong.com/spring-boot/spring-boot-hello-world-example-jsp/)
2. [springboottutorial](http://www.springboottutorial.com/creating-web-application-with-spring-boot)

# [Is web.xml required to deploy a spring boot application](https://stackoverflow.com/questions/36315591/is-web-xml-required-to-deploy-a-spring-boot-application)

<https://stackoverflow.com/questions/36315591/is-web-xml-required-to-deploy-a-spring-boot-application>

Q: I was trying to package a spring boot application as a war. According to [this](https://docs.spring.io/spring-boot/docs/current/reference/html/howto-traditional-deployment.html), I modified my Application class:

@SpringBootApplication

@EntityScan({"org.mdacc.rists.cghub.model"})

@EnableJpaRepositories(basePackages = {"org.mdacc.rists.cghub.ws.repository"})

public class Application extends SpringBootServletInitializer

{

public static void main( String[] args )

{

SpringApplication.run(Application.class, args);

}

@Override

protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {

return application.sources(Application.class);

}

}

Also added the following in my pom.xml

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-tomcat</artifactId>

<scope>provided</scope>

</dependency>

When I package the project though, I got the following error:

[ERROR] Failed to execute goal org.apache.maven.plugins:maven-war-plugin:2.2:war (default-war) on project cg-web: Error assembling WAR: webxml attribute is required (or pre-existing WEB-INF/web.xml if executing in update mode) -> [Help 1]

As I was reading upon spring boot application, I never saw anything about creating a web.xml. Is web.xml required in deploying spring boot application as war?

A: Do you have the web dependencies.

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

You can always have your web.xml if you need some configuration on it, just place the file in the correct folder within WEB-INF so spring can take it an read the configurations. Also change the packaging to

<packaging>war</packaging>

Consider as well use the parent pom for spring-boot as

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.3.2.RELEASE</version>

</parent>

This configuration

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-tomcat</artifactId>

<scope>provided</scope>

</dependency>

Only tell maven not to include tomcat dependencies in the war file, to avoid interference with the servlet provider.