1. Create a Spring Boot project using either IntelliJ's Spring Intializr or https://start.spring.io/
    a. Choose the packaging as WAR
    b. Add the Spring Web dependency so we don't have to do it later
2. In the generated project, go to pom.xml, replace the dependencies block with the one I provided in my example
    a. Specifically you need to add the tomcat-embedded-jasper dependency
    b. For the JSP examples to correctly work you also need the jstl dependency
3. Under the /project-name/src/main/ directory create a webapp directory
    a. Inside the webapp directory create a WEB-INF directory
        i. The actual folder to hold the jsp templates can vary, for my demo I created a jsp directory under WEB-INF to hold my templates
4. Make sure to put all your Java code under the same package as the class that contains the @SpringBootApplication annotation. In my example it's the com.example.springmvcdemo package. The @SpringBootApplication annotation includes the functionalities of several different annotations, one of which is @ComponentScan, which by default scans the package of the @SpringBootApplication class.
5. Create a view resolver that points to the directory that holds your jsp files
    a. You can copy my WebMvcConfig example
6. Now for testing you can create my HelloController and the helloWorld.jsp.
    a. Run your Spring Boot application
    b. Go to localhost:8080/hello-world in your browser and see if you can see the "Hello World!"