

# FAQ: How to configure the Spring Dispatcher Servlet using all Java Code (no xml)

## Question:

How to configure the Spring Dispatcher Servlet using all Java Code (no xml)?

## Answer:

Good question!

For Spring MVC, we cover all Java config (no xml) later in the course, complete with videos/explanation and everything.

However, if you just need the code, here are the steps

1. Delete the files: web.xml file and spring-mvc-demo-servlet.xml files
2. Create a new Java package: `com.luv2code.springdemo.config`
3. Add the following Java files in your package

## File: DemoAppConfig.java

```
1. package com.luv2code.springdemo.config;
2.
3. import org.springframework.context.annotation.Bean;
4. import org.springframework.context.annotation.ComponentScan;
5. import org.springframework.context.annotation.Configuration;
6. import org.springframework.web.servlet.ViewResolver;
7. import org.springframework.web.servlet.config.annotation.EnableWebMvc;
8. import org.springframework.web.servlet.view.InternalResourceViewResolver;
9.
10. @Configuration
11. @EnableWebMvc
12. @ComponentScan(basePackages="com.luv2code.springdemo")
13. public class DemoAppConfig {
14.
15.     // define a bean for ViewResolver
16. }
```

```

17.         @Bean
18.         public ViewResolver viewResolver() {
19.
20.             InternalResourceViewResolver viewResolver = new InternalResourceView
Resolver();
21.
22.             viewResolver.setPrefix("/WEB-INF/view/");
23.             viewResolver.setSuffix(".jsp");
24.
25.             return viewResolver;
26.         }
27.
28. }

```

---

## File: MySpringMvcDispatcherServletInitializer.java

```

1. package com.luv2code.springdemo.config;
2.
3. import org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;
4.
5. public class MySpringMvcDispatcherServletInitializer extends AbstractAnnotationConfigDispatcherServletInitializer {
6.
7.     @Override
8.     protected Class<?>[] getRootConfigClasses() {
9.         // TODO Auto-generated method stub
10.        return null;
11.    }
12.
13.    @Override
14.    protected Class<?>[] getServletConfigClasses() {
15.        return new Class[] { DemoAppConfig.class };
16.    }
17.
18.    @Override
19.    protected String[] getServletMappings() {
20.        return new String[] { "/" };
21.    }
22.
23. }

```

## 4. Test your app

Your app should work as desired.

---

I also uploaded a full project implementation with code here

[https://drive.google.com/open?id=1\\_5\\_2SggzgFHt7Rs2YYsv5JHRVX5Orq3](https://drive.google.com/open?id=1_5_2SggzgFHt7Rs2YYsv5JHRVX5Orq3)

---

For a discussion on how this code works, you can skip ahead to the following video

### **Video 403 - Spring MVC All Java Config**

<https://www.udemy.com/spring-hibernate-tutorial/learn/v4/t/lecture/8355870?start=148>

Skip ahead to time marker [02:28]

---

let me know if you need anything else.

:-)