



what is the role of

web.xml

Can be called deployment descriptor
First file loads when application started
Loads dispatcher

--> Java

dispatcher-servlet.xml

Can be called as front controller
Loads info about
controllers location
view location/type of views
view relover

--> Java

java configuration for web.xml

AbstractAnnotationConfigDispatcherServletInitializer
WebAppInitializer.java

getRootConfigClasses()
used to add additional configurations
ex.SessionFactory/EamilService....

getServletConfigClasses()
dispatcher config WebConfig.class

getServletMappings()
to define initial mapping '/'

```
class WebAppInitializer extends AbstractAnnotationConfigDispatcherServletInitializer
{
    Class[ ] getRootConfigClasses() { }

    Class[ ] getServletConfigClasses() { }

    String[ ] getServletMappings() { }
}
```

java configuration for dispatcher-servlet.xml WebConfig.java

```
@Configuration                      loads this file automatically
@ComponentScan(basePackages = "ord.example.controllers")
@EnableWebMvc
class WebConfig {

    create bean for InternalResourceViewResolver
    prefix
    suffix

}
```

```

1  <!DOCTYPE web-app PUBLIC
2      "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
3      "http://java.sun.com/dtd/web-app_2_3.dtd" >
4
5  <web-app>
6      <display-name>Archetype Created Web Application</display-name>
7
8      <!-- define dispatcher servlet info -->
9
10     <servlet>
11         <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
12         <servlet-name>dispatcher</servlet-name>
13     </servlet>
14
15     <servlet-mapping>
16         <servlet-name>dispatcher</servlet-name>
17         <url-pattern>/</url-pattern>
18         <!-- any request after / in BASE URL, request has to be processed as per dispatcher servlet config -->
19     </servlet-mapping>
20
21 </web-app>

```

web.xml
WebAppInitializer.java

```

WebAppInitializer.java
1  package org.example.config;
2
3  import org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;
4
5  public class WebAppInitializer extends AbstractAnnotationConfigDispatcherServletInitializer {
6
7
8      protected Class<?>[] getRootConfigClasses() {
9          return new Class[0];
10     }
11
12     protected Class<?>[] getServletConfigClasses() {
13         return new Class[]{WebConfig.class};
14     }
15
16     protected String[] getServletMappings() { return new String[]{"/"}; }
17
18 }

```

```

1 <beans xmlns = "http://www.springframework.org/schema/beans"
2     xmlns:context = "http://www.springframework.org/schema/context"
3     xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation = "http://www.springframework.org/schema/beans
5 http://www.springframework.org/schema/beans/spring-beans.xsd
6 http://www.springframework.org/schema/context
7 http://www.springframework.org/schema/context/spring-context.xsd">
8
9     <!-- location of controllers -->
10    <context:component-scan base-package="org.example.controllers"/>
11
12
13    <bean id="jspViewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">
14        <!-- which class to be used to resolve views from JSPs -->
15        <!-- location of views -->
16        <property name="viewClass" value="org.springframework.web.servlet.view.JstlView"/>
17        <property name="prefix" value="/WEB-INF/views/" />
18        <property name="suffix" value=".jsp" />
19    </bean>
20
21 </beans>

```

dispatcher-servlet.xml
WebConfig.java

```

1 package org.example.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 1 usage
11 @Configuration
12 @ComponentScan(basePackages = "org.example.controller")
13 @EnableWebMvc
14 public class WebConfig {
15     @Bean
16     public ViewResolver getViewResolver() { return new InternalResourceViewResolver( prefix: "/WEB-INF/views/", suffix: ".jsp"); }
17 }

```

```
main()
int a [ ] = {1,2,3,4,5,6,7};
display(a);

display( new int[ ] { 4,5,6,7,8,9} );
```

```
void display( int [ ] data)
{
for(int i: data)
    sop(i);
}
```