

在一个基于spring mvc的web应用中，需要有一个web容器，一般都是用tomcat，这个web容器提供了一个全局的上下文环境servletContext，同时也是Spring IOC容器的宿主环境。宿主容器启动时，其入口文件便是web.xml。web.xml需要定义两个最基本的内容，ContextLoaderListener和DispatchServlet。

1. ContextLoaderListener是实现了ServletContextListener接口的监听器，在项目启动的时候会触发容器初始化方法contextInitialized，容器初始化方法主要做3个事情。

1.1 执行initWebApplicationContext方法，启动一个根上下文

WebApplicationContext（一个接口类），其实现类为XmlWebApplicationContext（即传说中的IOC容器）

```
1 <!-- 定义监听器，监听项目的启动和关闭，执行对应的方法 -->
2 <listener>
3     <listener-class>org.springframework.web.context.ContextLoaderListener</l
4 </listener>
```

1.2根据配置路径加载spring文件创建bean的实例，这些bean都放入XmlWebApplicationContext容器中。

```
1 <!-- 加载Spring配置文件 -->
2 <context-param>
3     <param-name>contextConfigLocation</param-name>
4     <param-value>
5     classpath:spring-context*.xml
6     </param-value>
7 </context-param>
```

1.3 将加载好bean实例的WebApplicationContext按指定的key注册到ServletContext（java web容器变量宿主容器）容器中去/

2. contextLoaderListener监听器监听容器初始化完成之后，开始初始化web.xml中配置的Servlet（可以配置多个），springmvc中以DispatcherServlet为例。DispatcherServlet，前端控制器，也可以叫请求分发控制器，用来接收、匹配、处理、转发每个servlet请求。DispatcherServlet会以WebApplicationServlet作为自己的parent上下文，然后再出实话自己的上下文。DispatcherServlet出实话上下文主要包括处理器映射器，处理器适配器，视图解析器等。最后再将初始化完成的Servlet注册到ServletContext中，这样每个Servlet都可以持有自己的上下文，同时又共享根上下文，即IOC容器上下文WebApplicationContext。

```
1 <servlet>
2     <servlet-name>spring3mvc</servlet-name>
3     <servlet-class>org.springframework.web.servlet.DispatcherServlet</servle
4     <init-param>
5         <param-name>contextConfigLocation</param-name>
6         <param-value>
7             classpath:spring-mvc*.xml
8         </param-value>
9     </init-param>
10    <load-on-startup>1</load-on-startup>
11 </servlet>
```