**Spring 如何在 WEB 应用中使用 ?**

**1). 需要额外加入的 jar 包:**

spring-web-4.0.0.RELEASE.jar

spring-webmvc-4.0.0.RELEASE.jar

**2). Spring 的配置文件, 没有什么不同**

**3). 如何创建 IOC 容器 ?**

①. 非 WEB 应用在 main 方法中直接创建

②. 应该在 WEB 应用被服务器加载时就创建 IOC 容器:

在 ServletContextListener#contextInitialized(ServletContextEvent sce) 方法中创建 IOC 容器.

③. 在 WEB 应用的其他组件中如何来访问 IOC 容器呢 ?

在 ServletContextListener#contextInitialized(ServletContextEvent sce) 方法中创建 IOC 容器后, 可以把其放在ServletContext(即 application 域)的一个属性中.

④. 实际上, Spring 配置文件的名字和位置应该也是可配置的! 将其配置到当前 WEB 应用的初始化参数中较为合适.

**4). 在 WEB 环境下使用 Spring**

①. 需要额外加入的 jar 包:

spring-web-4.0.0.RELEASE.jar

spring-webmvc-4.0.0.RELEASE.jar

②. Spring 的配置文件, 和非 WEB 环境没有什么不同

**③. 需要在 web.xml 文件中加入如下配置:**

**Alt+/ 倒数有一个contextloaderlistener，选择，会自动出现，只需要修改黄标的部分为配置文件的路径即可**

<!-- 配置 Spring 配置文件的名称和位置 -->

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>**classpath:applicationContext.xml**</param-value>

</context-param>

<!-- 启动 IOC 容器的 ServletContextListener -->

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

**使用**

从Context域中获取获取一个applicationContext的IOC容器

**ApplicationContext ctx = WebApplicationContextUtils.**

**getWebApplicationContext(application);**

**利用WebApplicationContextUtils工具类的getWebApplicationContext方法，传入application对象就可以自动获取ioc容器**