**Web上下文剖析**

**ContextLoaderListener和DispatcherServlet的区别**

对于一个web容器，web容器提供了一个全局的上下文环境，这个上下文就是ServletContext，其为后面Spring IOC容器提供宿主环境。

在web容器启动时会触发容器初始化事件，ContextLoaderListener监听到这个事件后其ContextInitialized方法就会被调用，在这个方法中，spring会初始化一个启动上下文，这个上下文就是根上下文，也就是WebApplicationContext，实际实现类一般是XmlWebApplicationContext，这个其实就是spring的IoC容器，这个IoC容器初始化完后，Spring会将它存储到ServletContext，可供后面获取到该IOC容器中的bean。

在contextLoaderListener监听器初始化完毕后，开始初始化web.xml中配置的Servlet，这个servlet可以配置多个，以DispatcherServlet为例，这个servlet实际上是一个标准的前端控制器，用以转发、处理每个servlet请求。DispatcherServlet上下文在初始化的时候会建立自己的IoC上下文，用以持有spring mvc相关的bean。在建立DispatcherServlet自己的IoC上下文时，先从ServletContext中获取之前的根上下文(即WebApplicationContext)作为自己上下文的parent上下文。有了这个parent上下文之后，再初始化自己持有的上下文。这个servlet自己持有的上下文默认实现类也是XmlWebApplicationContext。初始化完毕后，spring以与servlet的名字相关的属性为Key，也将其存到ServletContext中。这样每个servlet就持有自己的上下文，即拥有自己独立的bean空间，同时各个servlet共享相同的bean，即根上下文(WebApplicationContext)。

总结：1、ContextLoaderListener中创建ApplicationContext主要用于整个Web应用程序需要共享的一些组件，比如DAO，[数据库](http://lib.csdn.net/base/mysql" \t "https://www.cnblogs.com/weknow619/p/_blank)的ConnectionFactory等。而由DispatcherServlet创建ApplicationContext主要用于和该Servlet相关的一些组件，比如Controller、ViewResovler等。

2、对于作用范围而言，在DispatcherServlet中可以引用由ContextLoaderListener所创建的ApplicationContext，而反过来不行。