InitBeanWraper(bw)默认实现为空.

# ApplicationContext

为应用程序配置提供中央接口，当应用程序运行时，这是只读的，但是它可能重新加载，如果这个实现支持这一点。

应用程序上下文提供了：

1. Bean工厂方法用于访问应用程序组件，从ListableBeanFacoty继承。

# HttpServletBean

Httpservlet的简单扩展，他将其配置参数（web.xml中init-param参数）视为bean属性，对任何类型的servlet他都是方便的父类，配置参数类型转换是自动的，转换后的值可以调用setter方法。子类可以指定请求属性，没有匹配bean属性的参数将要被忽略。

这个servlet将请求交给子类处理，默认继承了了httpservlet(doGet,doPost,etc)。

这个servlet基类没有依赖ApplicationContext概念，简单的servlet通常不加载自己的context,而是从spring root应用程序上下文中访问服务bean.

FrameworkServlet类是一个更详细的servlet基类，它可以加载自己的应用程序的上下文

# FrameworkServlet

Spring web框架基本Servlet,提供与Spring应用程序上下文的集成，在一个JavaBean-based全部解决方案中。

这个类提供类一下功能：1.管理每一个servlet的WebApplicationContext的实例，在servlet的命名空间中的bean决定着servlet的配置。

2.发布事件在请求处理上，无论请求是否成功处理。

子类必须实现doService方法处理请求，因为这是直接继承了HttpServletBean而不是HttpServlet，bean的属性会自动匹配。子类可以重写initFrameworkServlet方法，为了自定义初始化。

在servlet的init-param的层次上检测contextClass参数，回调默认的上下文类，XmlWebApplicationContext,如果没有发现，注意，有了默认的FrameworkServlet，一个自定义的上下文类必须要实现ConfigurableWebApplicationContext SPI。

接受一个可选的”contextInitializerClasses”servlet init-param，它指出一个或多个ApplicationContextInitializer类，管理web应用程序上下文委托初始化器，允许另外编程式的配置文件，例如，添加属性源或者激活的配置文件对在上下文环境中。参见ContextLoader,它支持一个”contextInitializerClassess”的context-param，它具有相同的语义对于”root”web应用程序上下文。

传递一个”contextInitializerClasses”servlet 初始化参数到上下文实例中，将其解析成多个文件路径，这些路径可以有任何数量的逗号和空格分隔，像”test-servlet.xml,myServlet.xml”，如果没有明确指定，上下文实现应该从servlet的命名空间构建缺省位置。

注意：在多个配置文件下，稍后定义的bean将要重写加载文件较早定义的bean，至少在使用Spring默认的ApplicatonContext的实现。这可以通过一个额外的XML文件故意覆盖某些bean的定义。

默认的命名是”servlet-name”-servlet,例如，”test-servelt”对一个servlet的名字是test，也可以通过”namespace”servlet inti-param显示地设置命名空间。

# WebApplicationContext

为web应用程序提供配置的接口，当应用程序运行时，这是只读的，但是如果实现支持这一点，它可以会被重新加载。这个接口添加了getServeltContext()方法在通用的应用程序上下文接口中，并且定义了一个著名的应用程序属性名，root上下文必须在bootstrap过程中绑定它。像通用的应用程序上下文，web应用程序上下文是分层的，每一个应用程序都只有一个根上下文，在应用程序中每一个servlet(包括MVC框架中的dispatcher servlet)都有自己的子上下文。除了标准的应用程序上下文生命周期功能之外，webApplicationContext实现需要检测ServletContextAware bean并且调用相应的setServletContext方法。

## ConfigurableWebApplicationContext

通过可配置的web应用程序上下文实现的接口。由ContextLoader和FrameworkServlet支持。

注意：在调用从ConfigurableApplicationContext继承的ConfigurableApplicationContext.refresh()方法之前，需要调用这个接口的setter方法，他们不会

## WebApplicationContextUtils

为给定的ServeltContext检索根WebApplicationContext的便利方法，这对于已编程方式从自定义web视图或MVC动作中访问Spring应用程序上下文非常有用。

注意：对于许多的web框架，有许多的便利方式访问根上下文，无论作为Spring的一部分，或者作为外部可用的库。这个助手类这是访问树上下文的最通用的方式。