在 Servlet API 中有一个 ServletContextListener 接口，它能够监听 ServletContext 对象的生命周期，实际上就是监听 Web 应用的生命周期。

当Servlet 容器启动或终止Web 应用时，会触发ServletContextEvent 事件，该事件由 ServletContextListener 来处理。在 ServletContextListener 接口中定义了处理ServletContextEvent 事件的两个方法。

contextInitialized(ServletContextEvent sce) ：当Servlet 容器启动Web 应用时调用该方法。在调用完该方法之后，容器再对Filter 初始化，并且对那些在Web 应用启动时就需要被初始化的Servlet 进行初始化。

contextDestroyed(ServletContextEvent sce) ：当Servlet 容器终止Web 应用时调用该方法。在调用该方法之前，容器会先销毁所有的Servlet 和Filter 过滤器。

下面通过两个具体的例子来介绍 ServletContextListener 的用法。

例一：在服务启动时，将数据库中的数据加载进内存，并将其赋值给一个属性名，其它的 Servlet 就可以通过 getAttribute 进行属性值的访问。有如下两个步骤：

1 ： ServletContext 对象是一个为整个 web 应用提供共享的内存，任何请求都可以访问里面的内容

2 ：如何实现在服务启动的时候就动态的加入到里面的内容：我们需要做的有：

1 ） 实现 servletContextListerner 接口 并将要共享的通过 setAttribute （ name,data ）方法提交到内存中去   ；

2 ）应用项目通过 getAttribute(name) 将数据取到 。

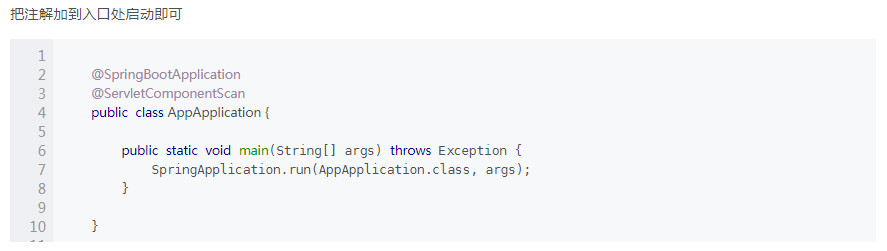
## 方案一

方案一采用原生Servlet3.0的注解进行配置、@WebServlet 、@WebListener、@WebFilter是Servlet3.0 api中提供的注解   
通过注解可以完全代替web.xml中的配置，下面是一个简单的配置









## 方案二

方案二是采用自己SpringBoot 配置bean的方式进行配置的，SpringBoot提供了三种BeanFilterRegistrationBean、ServletRegistrationBean、ServletListenerRegistrationBean

分别对应配置原生的Filter、Servlet、Listener,下面提供的三个配置和方案一采用的方式能够达到统一的效果



## 总结

两种方案在使用上有差别，但是在内部SpringBoot的实现上是无差别的，即使使用的是Servlet3.0注解，也是通过扫描注解

转换成这三种bean的FilterRegistrationBean、ServletRegistrationBean、ServletListenerRegistrationBean