# SpringBoot初始教程之Servlet、Filter、Listener配置(七)

本文链接：<https://blog.csdn.net/veZunShao/article/details/53116744>

# 1.介绍

通过之前的文章来看，SpringBoot涵盖了很多配置，但是往往一些配置是采用原生的Servlet进行的，但是在SpringBoot中不需要配置web.xml的

因为有可能打包之后是一个jar包的形式，这种情况下如何解决？SpringBoot 提供了两种方案进行解决。

# 2.快速开始

## 2.1 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。

### IndexServlet.

@WebServlet(name = "IndexServlet",urlPatterns = "/hello")

public class IndexServlet extends HttpServlet {

@Override

public void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

resp.getWriter().print("hello word");

resp.getWriter().flush();

resp.getWriter().close();

}

@Override

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

this.doGet(req, resp);

}

}

### IndexListener.

@WebListener

public class IndexListener implements ServletContextListener {

private Log log = LogFactory.getLog(IndexListener.class);

@Override

public void contextInitialized(ServletContextEvent servletContextEvent) {

log.info("IndexListener contextInitialized");

}

@Override

public void contextDestroyed(ServletContextEvent servletContextEvent) {

}

}

### IndexFilter.

@WebFilter(urlPatterns = "/\*", filterName = "indexFilter")

public class IndexFilter implements Filter {

Log log = LogFactory.getLog(IndexFilter.class);

@Override

public void init(FilterConfig filterConfig) throws ServletException {

log.info("init IndexFilter");

}

@Override

public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, FilterChain filterChain) throws IOException, ServletException {

log.info("doFilter IndexFilter");

filterChain.doFilter(servletRequest,servletResponse);

}

@Override

public void destroy() {

}

}

上面配置完了，需要配置一个核心的注解@ServletComponentScan,具体配置项如下，可以配置扫描的路径

@Target(ElementType.TYPE)

@Retention(RetentionPolicy.RUNTIME)

@Documented

@Import(ServletComponentScanRegistrar.class)

public @interface ServletComponentScan {

@AliasFor("basePackages")

String[] value() default {};

@AliasFor("value")

String[] basePackages() default {};

Class<?>[] basePackageClasses() default {};

}

把注解加到入口处启动即可

@SpringBootApplication

@ServletComponentScan

public class AppApplication {

public static void main(String[] args) throws Exception {

SpringApplication.run(AppApplication.class, args);

}

}

## 2.2 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。

### ServletRegistrationBean .

@Bean

public ServletRegistrationBean indexServletRegistration() {

ServletRegistrationBean registration = new ServletRegistrationBean(new IndexServlet());

registration.addUrlMappings("/hello");

return registration;

}

### FilterRegistrationBean.

@Bean

public FilterRegistrationBean indexFilterRegistration() {

FilterRegistrationBean registration = new FilterRegistrationBean(new IndexFilter());

registration.addUrlPatterns("/");

return registration;

}

ServletListenerRegistrationBean.

@Bean

public ServletListenerRegistrationBean servletListenerRegistrationBean(){

ServletListenerRegistrationBean servletListenerRegistrationBean = new ServletListenerRegistrationBean();

servletListenerRegistrationBean.setListener(new IndexListener());

return servletListenerRegistrationBean;

}

# 3.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。

# 4.扩展

大家在使用的时候有没有发觉，其实SpringBoot在使用SpringMvc的时候不需要配置DispatcherServlet的，因为已经自动配置了，

但是如果想要加一些初始配置参数如何解决，方案如下：

@Bean

public ServletRegistrationBean dispatcherRegistration(DispatcherServlet dispatcherServlet) {

ServletRegistrationBean registration = new ServletRegistrationBean(dispatcherServlet);

registration.addUrlMappings("\*.do");

registration.addUrlMappings("\*.json");

return registration;

}

可以通过注入DispatcherServlet 然后用ServletRegistrationBean包裹一层 动态的加上一些初始参数。