



## 《Spring Boot 基础教程》

## 第 10 节

## web 应用开发-Servlets, Filters, listeners

一. Web 开发使用 Controller 基本上可以完成大部分需求，但是我们还可能会用到 Servlet、Filter、Listener 等等

二. 在 spring boot 中的三种实现方式

方法一：通过注册 ServletRegistrationBean、FilterRegistrationBean 和 ServletListenerRegistrationBean 获得控制

```
/**
 * 自定义 servlet
 *
 * @author wujing
 */
public class CustomServlet extends HttpServlet {

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        System.out.println("servlet get method");
        doPost(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        System.out.println("servlet post method");
        response.getWriter().write("hello world");
    }
}

/**
 * 自定义 filter
 *
 * @author wujing
 */
```



```
public class CustomFilter implements Filter {

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
        System.out.println("init filter");
    }

    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
        throws IOException, ServletException {
        System.out.println("do filter");
        chain.doFilter(request, response);
    }

    @Override
    public void destroy() {
        System.out.println("destroy filter");
    }

}

/**
 * 自定义 listener
 *
 * @author wujing
 */
public class CustomListener implements ServletContextListener {

    @Override
    public void contextInitialized(ServletContextEvent sce) {
        System.out.println("contextInitialized");
    }

    @Override
    public void contextDestroyed(ServletContextEvent sce) {
        System.out.println("contextDestroyed");
    }

}

注册 bean

@Bean
public ServletRegistrationBean servletRegistrationBean() {
```



```
return new ServletRegistrationBean(new CustomServlet(), "/roncoo");
}

@Bean
public FilterRegistrationBean filterRegistrationBean() {
    return new FilterRegistrationBean(new CustomFilter(), servletRegistrationBean());
}

@Bean
public ServletListenerRegistrationBean<CustomListener> servletListenerRegistrationBean() {
    return new ServletListenerRegistrationBean<CustomListener>(new CustomListener());
}
```

方法二：通过实现 ServletContextInitializer 接口直接注册

```
implements ServletContextInitializer {

    @Override
    public void onStartUp(ServletContext servletContext) throws
    ServletException {
        servletContext.addServlet("customServlet", new
        CustomServlet()).addMapping("/roncoo");
        servletContext.addFilter("customFilter", new
        CustomFilter())

        .addMappingForServletNames(EnumSet.of(DispatcherType.REQUEST), true, "customServlet");
        servletContext.addListener(new CustomListener());
    }
}
```

方法三：在 SpringBootApplication 上使用 @ServletComponentScan 注解后，直接通过 @WebServlet、@WebFilter、@WebListener 注解自动注册

更多课程信息，请关注 龙果学院 官方网站 <http://www.roncoo.com/>

或关注 龙果 微信公众号 RonCoo\_com

