



链滴

# springBoot 配置 filter 的两种方式

作者: [luomuren](#)

原文链接: <https://ld246.com/article/1648015560688>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# springBoot配置filter的两种方式

## 一、使用注解

### 1. 假设Filter类的路径为com.sanro.filter

```
@Slf4j
@WebFilter(filterName = "authFilter", urlPatterns = "/*")
@Order(1) //测试好像这个参数不生效，实际生效的是Filter扫描到的顺序（所以起名很重要）
public class AuthFilter implements Filter {
    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletException {
        log.debug("start to auth request validate...111");
        HttpServletRequest req = (HttpServletRequest) request;
        String token = req.getHeader("token");
        if (token != null) {
            // :TODO check token
            log.info("auth success");
            chain.doFilter(request, response);
        } else {
            log.err("auth failed");
        }
    }
}
```

### 2.自动扫描配置如下

```
@Slf4j
@EnableScheduling
@ComponentScan(basePackages = {"com.sanro.filter"}) //这一句完成了配置，Springboot的“懒理念”真的厉害。
@SpringBootApplication
public class Application {
    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
        log.debug("rest server start success.");
    }
}
```

## 二、手动配置

### 1.Filter类

```
@Slf4j
@Component
public class AuthFilter implements Filter {
    private static final Logger log = LogManager.getLogger(ClientFilter.class);
    private static String value = "";
}
```

```

@Override
public void init(FilterConfig filterConfig) throws ServletException
{
    value = filterConfig.getInitParameter("value");
}

@Override
public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletException {
    log.debug("start to auth request validate...111");
    log.info(value);
    HttpServletRequest req = (HttpServletRequest) request;
    String token = req.getHeader("token");
    if (token != null) {
        // :TODO check token
        log.info("auth success");
        chain.doFilter(request, response);
    } else {
        log.err("auth failed");
    }
}
}
}

```

filter中无法直接获取properties配置文件中的值，可以通过Filter配置类传入

## 2.Filter配置类

```

@Configuration
public class FilterConfig {

    @Value("${value}")
    private String value;

    @Bean
    public FilterRegistrationBean registerAuthFilter() {
        FilterRegistrationBean registration = new FilterRegistrationBean();
        registration.setFilter(authFilter);
        registration.addUrlPatterns("/*");
        registration.setName("authFilter");
        registration.setOrder(1); //值越小，Filter越靠前。
        // 传入参数
        Map<String, String> initParameters = new HashMap<String, String>();
        initParameters.put("value", value);
        registration.setInitParameters(initParameters);
        return registration;
    }

    //如果有多个Filter，再写一个public FilterRegistrationBean registerOtherFilter(){...}即可。
}

```

## 版权声明

作者：[一沙世界](#)

地址: <https://www.cnblogs.com/yoyotl/p/10853499.html>