

SpringBoot 基于注解的 AOP

作者: [tiangao](#)

原文链接: <https://ld246.com/article/1531107136259>

来源网站: [链滴](#)

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

<blockquote>

<p>google 搜索 spring aop annotation, 看了一大堆, 好多过时的内容, 要么就特别复杂, 再搜文的 “spring 注解 代理”, 看了半天也没啥帮助, 今天试着试着突然搞定了, 特别简单</p>

</blockquote>

<p>springboot 用不着配置文件</p>

<h2 id="定义注解">定义注解</h2>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">@Target(ElementType.METHOD)
</span></span><span class="highlight-line"><span class="highlight-cl">@Retention(Reten
ionPolicy.RUNTIME)
</span></span><span class="highlight-line"><span class="highlight-cl">public @interface
MapCache {
</span></span><span class="highlight-line"><span class="highlight-cl">}</span></span></code></pre>
```

<h2 id="配置后置处理器">配置后置处理器</h2>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">@Configuration
</span></span><span class="highlight-line"><span class="highlight-cl">public class AopC
nfig implements BeanPostProcessor {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    @Override
</span></span><span class="highlight-line"><span class="highlight-cl">    public Object po
tProcessBeforeInitialization(Object bean, String s) throws BeansException {
</span></span><span class="highlight-line"><span class="highlight-cl">        return bean;
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    @Override
</span></span><span class="highlight-line"><span class="highlight-cl">    public Object po
tProcessAfterInitialization(Object bean, String beanName) throws BeansException {
</span></span><span class="highlight-line"><span class="highlight-cl">        //这个地方不
加过滤会导致 tomcat 启动失败, 目前还没定位到确切原因
</span></span><span class="highlight-line"><span class="highlight-cl">        if (!bean.getC
lass().getName().startsWith("com.example.wsdemo")) {
</span></span><span class="highlight-line"><span class="highlight-cl">            return bea
;
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">        ProxyFactory
factory = new ProxyFactory();
</span></span><span class="highlight-line"><span class="highlight-cl">        factory.setTa
get(bean);
</span></span><span class="highlight-line"><span class="highlight-cl">        factory.addA
dvisor(new MapCacheAdvice());
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">        return factory
getProxy();
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">}</span></span></code></pre>
```

<p><code>MapCacheAdvice</code> 源码</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">public class MapCacheAdvice extends DefaultPointcutAdvisor {
</span></span><span class="highlight-line"><span class="highlight-cl">    @Override
</span></span><span class="highlight-line"><span class="highlight-cl">    public Pointcut
```

```

etPointcut() {
</span></span><span class="highlight-line"><span class="highlight-cl">    return new A
notationMatchingPointcut(null, MapCache.class);
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    @Override
</span></span><span class="highlight-line"><span class="highlight-cl">    public Advice ge
Advice() {
</span></span><span class="highlight-line"><span class="highlight-cl">        return new
apCacheInterceptor();
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    public static cla
s MapCacheInterceptor implements MethodInterceptor, Advice {
</span></span><span class="highlight-line"><span class="highlight-cl">        @Override
</span></span><span class="highlight-line"><span class="highlight-cl">    public Object in
oke(MethodInvocation invocation) throws Throwable {
</span></span><span class="highlight-line"><span class="highlight-cl">        System.out
println("=====");
</span></span><span class="highlight-line"><span class="highlight-cl">        Object res
lt = invocation.proceed();
</span></span><span class="highlight-line"><span class="highlight-cl">        System.out
println("-----");
</span></span><span class="highlight-line"><span class="highlight-cl">        return resu
t;
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl"></code></pre>
<h2 id="测试">测试</h2>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">@Service
</span></span><span class="highlight-line"><span class="highlight-cl">public class SaySe
vice {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    @Autowired
</span></span><span class="highlight-line"><span class="highlight-cl">    BBService bbSe
vice;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    public String sa
() {
</span></span><span class="highlight-line"><span class="highlight-cl">        System.out.pr
ntln("+++ say +++");
</span></span><span class="highlight-line"><span class="highlight-cl">        return bbServ
ce.getValue() + " hello";
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    @MapCache
</span></span><span class="highlight-line"><span class="highlight-cl">    public String sa
(String name) {
</span></span><span class="highlight-line"><span class="highlight-cl">        System.out.pr
ntln("+++ say hello +++");
</span></span><span class="highlight-line"><span class="highlight-cl">        return bbServ
ce.getValue() + " hello " \+ name;

```

```

}
}
}
}
@Controller
public class Index
ontroller {
    @ResponseBody
    @GetMapping(
/test")
    public String sa
() {
        sayService.sa
("pk");
        return saySer
ice.say();
    }
}
}
}
}

```

调用结果

```

=====
+++ say hello ++
-----
+++ say +++

```