

# tomcat 学习 | 源码学习方法及 tomcat 处理 Web 请求

作者: [xiaodaojava](#)

原文链接: <https://ld246.com/article/1565794532595>

来源网站: [链滴](#)

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

<p></p>

<h2 id="开头说两句">开头说两句</h2>

<p>小刀博客: <a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.lixiang.red" target="blank" rel="nofollow ugc">https://www.lixiang.red</a><br>

小刀公众号: 程序员学习大本营</p>

<h2 id="学习背景">学习背景</h2>

<p>上一波,我们一起新建了 web 应用,然后把 war 包丢到 tomcat 源码中的 webapps 目录,然后通源码启动的方式成功 debug 运行了我们 web 应用,并且在浏览器上看到了相应的输出:<br>

<a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.lixiang.red%2Farticles%2F019%2F08%2F13%2F1565705566068.html" target="\_blank" rel="nofollow ugc">https://www.lixiang.red/articles/2019/08/13/1565705566068.html</a></p>

<h2 id="源码学习方法">源码学习方法</h2>

<h2 id="能debug就debug">能 debug 就 debug</h2>

<p>直接看源码就很容易头晕,加上现在面向接口编程,我们很难直观看到在运行时,这个接口实际的实类是什么<br>

但在 debug 界面就不一样了,我们可以清楚的看到变量的状态值,调用的接口实际上是哪个实现类,如图所示<br>

<br>

所以说,学习源码,公司的项目代码,要运行起来,这样还可以边改边看效果</p>

<h2 id="寻找有效文档并分段">寻找有效文档并分段</h2>

<p>直接从代码看,如果不找文档的话,没有一个学习的入口,不知道从什么地方下手,面对庞大的代码,就感到学习不下,没法学<br>

我们最开始学习 tomcat 时,是因为我们知道平时启动 tomcat 是从 startup.sh 开始的,所以我们沿着个脚本,一步步的找到了 Bootstrap.java 的 main 函数,然后通过 main 函数又一步步的研究到各个组件<br>

从文档中找到学习的入口,可能是某个类的某个方法,可能是张流程图等等,如现在我们要去学习 tomcat 处理 web 请求,我们可以从相关书籍上找到,是从 CoyoteAdapter.service 进行的主要处理.这时候,我们就找到了我们学习的入口,然后打断点在方法入口处,后面就可以一步一步的跟踪下去了<br>

然后通过文档里面的关键点,找到代码的对应部分,然后划分代码段落,找到每段代码的大致意思</p>

<h2 id="建立知识库">建立知识库</h2>

<p>知识主要分已知和未知,我们大部分的学习,都应该是从已知到未的过程,随着我们学习的深入,不断有知识点被加入到我们的已知库里面.<br>

如:我们现在知道知识点 A,然后可以从 A 推导出 B,C.那么就把 B,C 也加入到我们的已知库里面,下次学习的时候,就可以用 B 推导 D, C 推导 E 等等,这样不断的去壮大.<br>

现在对于我们 tomcat 学习,我们的已知有以下几点:<br>

A. 我们已知请求的 url ,http://localhost:8088/study/hello<br>

B. 我们已知请求处理的方法入口: CoyoteAdapter.service<br>

C. 我们通过以前的文章学习已知: 有 coyote 组件, Connector 组件, Engine 组件 等等<br>

那么下面,我们开始边看断点的运行状态边推导,如下图所示<br>

由 A 和断点状态,可以看到我们的 url 最开始是和 <code>org.apache.coyote.Request</code> 是应起来的,<br>

然后用 Connector.createRequest 去创建了一个 <code>org.apache.catalina.connector</code> 下的 Request,也是我们后面一直用的 request.<br>

这个就是新知识 D, 我们把他加入到我们已知库里面, 然后类似于这样,不断的去壮大已知库<br>

</p>

<h2 id="web请求处理">web 请求处理</h2>

<p>通过上面的方法,我们可以运用到今天的源码学习中.<br>

通过查找资料,我们可以看到 tomcat 处理 web 请求时,有以下几个关键点:<br>

1.CoyoteAdapter.Service<br>

2.请求映射<br>

3.获取并执行 Engine 的 valve<br>

4. 判断是否是同步/异步 <br>

5. 执行对应的逻辑 <br>

我们对源码的分段如下图所示,这样本来 130 多行的源代码,我们只用近 10 行就把大致意思,过程给精出来了:<br>

 <br>

然后我们就可以深入到每一段,看他是怎么处理的,可以循环使用上面的学习方法 <br>

下面,我们以请求映射为例一起深入学习下 </p>

<h2 id="请求映射">请求映射 </h2>

<p>同样,我们先进行代码分段,如下图所示,可以看到 300 行的代码我们一个屏幕内就可以看到整个大的逻辑,后面我们可以再把这个再深入,然后分段,这样一直学习下去 <br>

 </p>

<h2 id="最后说两句">最后说两句 </h2>

<p>后面的学习,基本上都是重复这一段,不停的分段,然后不停去看小段里面的逻辑,直到最底层的,大家油! 在学习过程中有什么想法,可以随时和小刀分享: best396975802 </p>