

# 从jbossweb看Tomcat之一

2016-06-30 12:43 feiying 0 阅读 106

前面这段时间，我们深入分析了Tomcat的前端组件，这个阶段，我们以Jbossweb的视角来看看jboss是怎么封装Tomcat的。

## 1.jbossweb项目是什么

来看一段 <http://jbossweb.jboss.org/> 中的最开头的描述

What is JBoss Web

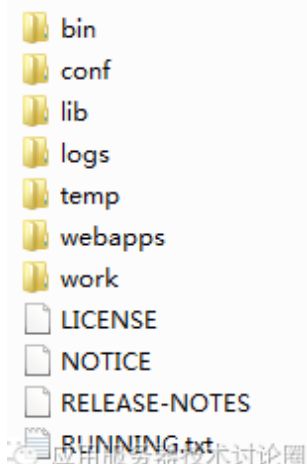
JBoss Web Server is an enterprise ready web server designed for medium and large applications, based on Tomcat.

JBoss Web a component of the JBoss Application Server, there are no more standalone version of JBoss Web you need the Application Server to get the Servlet/JSP container.,

JBoss Web Server provides organizations with a single deployment platform for Java Server Pages (JSP) and Java Servlet technologies, PHP, and CGI. It uses a genuine high performance hybrid technology that incorporates the best of the most recent OS technologies for processing high volume data, while keeping all the reference Java specifications.

分析得出，Jboss web实际就是以Tomcat为基础，集成到Jboss的web容器。

我们可以拿一个实际的下载Jboss web 2的目录进行分析：(Jbossweb 3已经发生了变化)



这其实实际就是Tomcat的整体目录，bin目录，conf目录没有什么区别，对于lib目录其实也是一样：

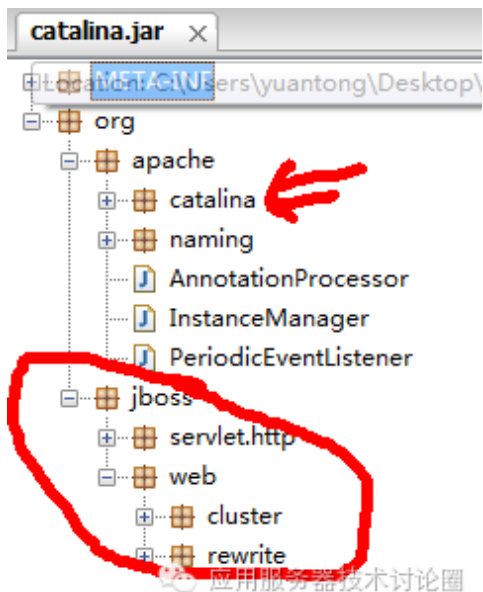
下载《开发者大全》

下载 (/download/dev.apk)



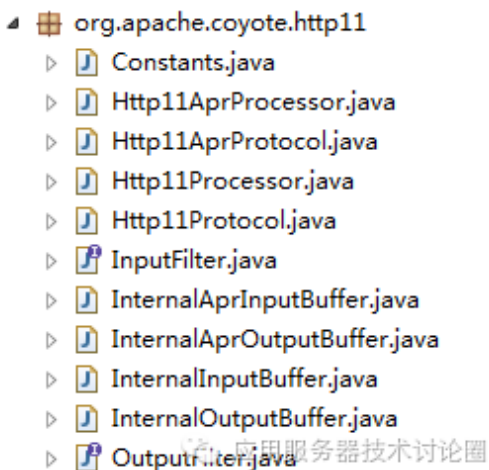


但是区别实际就在于catalina.jar包中就有Jboss的修改

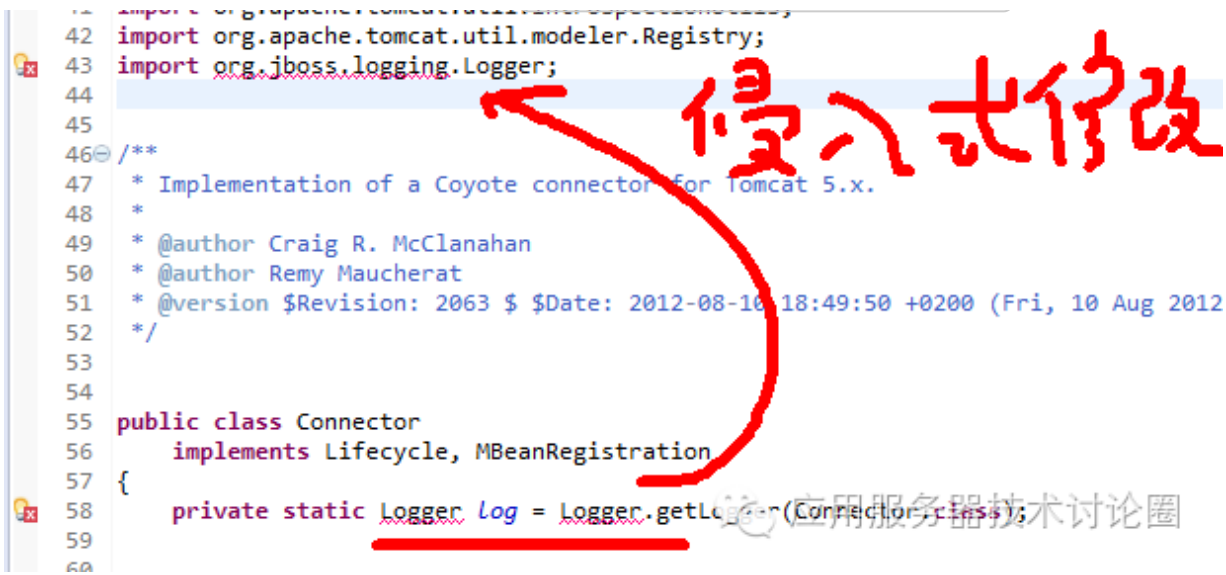


而对应的catalina包，也不是原来的对应tomcat的某个版本，以该版本为例，是基于tomcat6.0的早期版本来的做的；

分析一下org.apache.coyote.http11包



可以看到该包中连NIO通道都没有，而且随便找一个类，Connector：



```

42 import org.apache.tomcat.util.modeler.Registry;
43 import org.jboss.logging.Logger;
44
45 /**
46  * Implementation of a Coyote connector for Tomcat 5.x.
47  *
48  * @author Craig R. McClanahan
49  * @author Remy Mauchrat
50  * @version $Revision: 2063 $ $Date: 2012-08-10 18:49:50 +0200 (Fri, 10 Aug 2012)
51  */
52
53
54
55 public class Connector
56     implements Lifecycle, MBeanRegistration
57 {
58     private static Logger log = Logger.getLogger(Connector.class);
59
60

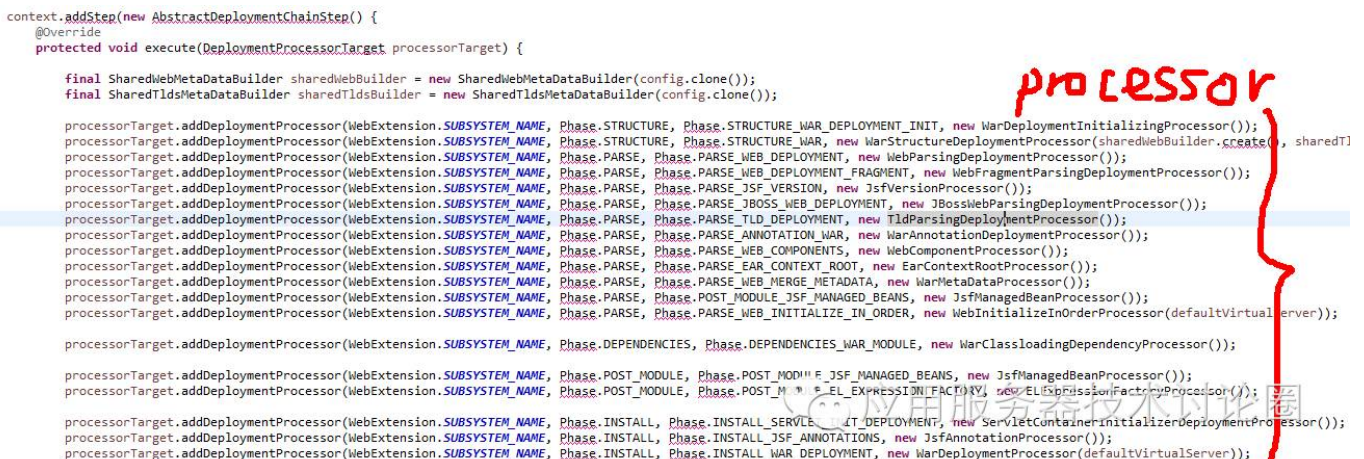
```

发现在catalina类也是侵入式的修改。

通过上面的分析，可以总结得出，jbossweb实际是基于Tomcat来做的，在Tomcat的某个版本中，加上jboss自己的特性。

而jbossweb的特点在于优化了性能，可以从几点来看：

- 基于jboss自己的log系统，优化了原有Tomcat的log
- 采用jboss-native，对接APR和Openssl，而彼时jvm效率较低，这种方法相当于让Tomcat前端能“享受”OS的最高级的优化特性
- 修改了lifecycle部分，优化了Tomcat组件启动和停止，减少了线程消耗
- 重新梳理了deploy的过程，在优化了原有的tomcat部署部分，加了很多ee特性：



```

context.addStep(new AbstractDeploymentChainStep() {
    @Override
    protected void execute(DeploymentProcessorTarget processorTarget) {
        final SharedWebMetaDataBuilder sharedWebBuilder = new SharedWebMetaDataBuilder(config.clone());
        final SharedTldsMetaDataBuilder sharedTldsBuilder = new SharedTldsMetaDataBuilder(config.clone());

        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.STRUCTURE, Phase.STRUCTURE_WAR_DEPLOYMENT_INIT, new WarDeploymentInitializingProcessor());
        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.STRUCTURE, Phase.STRUCTURE_WAR, new WarStructureDeploymentProcessor(sharedWebBuilder.create(), sharedTldsBuilder));
        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.PARSE, Phase.PARSE_WEB_DEPLOYMENT, new WebParsingDeploymentProcessor());
        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.PARSE, Phase.PARSE_WEB_DEPLOYMENT_FRAGMENT, new WebFragmentParsingDeploymentProcessor());
        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.PARSE, Phase.PARSE_JSF_VERSION, new JsVersionProcessor());
        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.PARSE, Phase.PARSE_JBOSS_WEB_DEPLOYMENT, new JBossWebParsingDeploymentProcessor());
        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.PARSE, Phase.PARSE_TLD_DEPLOYMENT, new TldParsingDeploymentProcessor());
        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.PARSE, Phase.PARSE_ANNOTATION_WAR, new WarAnnotationDeploymentProcessor());
        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.PARSE, Phase.PARSE_WEB_COMPONENTS, new WebComponentProcessor());
        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.PARSE, Phase.PARSE_EAR_CONTEXT_ROOT, new EarContextRootProcessor());
        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.PARSE, Phase.PARSE_WEB_MERGE_METADATA, new WarMetaDataProcessor());
        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.PARSE, Phase.POST_MODULE_JSF_MANAGED_BEANS, new JsManagedBeanProcessor());
        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.PARSE, Phase.PARSE_WEB_INITIALIZE_IN_ORDER, new WebInitializeInOrderProcessor(defaultVirtualServer));

        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.DEPENDENCIES, Phase.DEPENDENCIES_WAR_MODULE, new WarClassloadingDependencyProcessor());

        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.POST_MODULE, Phase.POST_MODULE_JSF_MANAGED_BEANS, new JsManagedBeanProcessor());
        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.POST_MODULE, Phase.POST_MODULE_JEL_EXPRESSION_FACTORY, new JELExpressionFactoryProcessor());

        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.INSTALL, Phase.INSTALL_SERVLET_DEPLOYMENT, new ServletContainerInitializerDeploymentProcessor());
        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.INSTALL, Phase.INSTALL_JSF_ANNOTATIONS, new JsAnnotationProcessor());
        processorTarget.addDeploymentProcessor(WebExtension.SUBSYSTEM_NAME, Phase.INSTALL, Phase.INSTALL_WAR_DEPLOYMENT, new WarDeploymentProcessor(defaultVirtualServer));
    }
});

```

值得一提的是，最新版本的JbossWeb，因为Jboss社区已经有了Wildfly，这个Wildfly是基于undertow的前端框架，相当于和Tomcat一个层级的Web容器，因而JbossWeb的大方向已经转向undertow的包装，并且在Wildfly的新版本中继续提供web请求访问的支撑。

而我们关注的视角是，以tomcat为基础，看看JbossWeb在Tomcat版本的修改和变化，对于新版本的Wildfly和undertow，在后续的篇章中会作为详细阐述的重点，不在这个系列之中。

下载《开发者大全》

下载 (/download/dev.apk)



## 2.jboss各种版本

因为Jboss的版本比较众多，这里提一下Jboss的各个版本。

a.Jboss Application Server，俗称Jboss AS，这个是最常见的，Jboss的JAVA EE应用服务器，是开源的版本

<http://jbossas.jboss.org/downloads.html> 这个链接就是 Jboss AS的老版本的下载

Version	Description	Release Date	License	Support	Download
JBoss AS 7.1.1.Final	AS Certified Java EE 6 Full Profile	2012-03-09	LGPL	No support, archived release!	ZIP (127MB) Release Notes
	AS Certified Java EE 6 Full Profile	2012-03-09	LGPL	No support, archived release!	TAR.GZ (127MB) Release Notes
	Quickstarts	2012-03-09	AL 2	No support, archived release!	site Release Notes
JBoss AS 7.1.0.Final	AS Certified Java EE 6 Full Profile	2012-02-16	LGPI	No support, archived release!	ZIP (100MB) Release Notes
	AS Certified Java EE 6 Full Profile	2012-02-16	LGPL	No support, archived release!	TAR.GZ (100MB) Release Notes

不过，对于该系列来讲，在Jboss AS 7之后，Jboss社区推出了Wildfly，改变了原有的Jboss的架构，地址在 <http://www.wildfly.org/>

这个Wildfly不单单对原来的Jboss整体架构的一次革新，而且对其web容器部分不再依赖于tomcat，转而自行研发了undertow，用以替代Tomcat的前端。

b.Jboss Enterprise Application Server, 俗称EAP

这个是企业级的版本，与Jboss AS相差不大，很多模块几乎是一样的，但在企业级的功能上做了一些微调。

下载的地址为：<http://www.jboss.org/products/eap/download/>

我们可以看看Jboss EAP 6的对应情况：

### JBoss Enterprise Application Platform 6

The following components have been integrated into a single distribution for JBoss Enterprise Application Platform 6. This integrated Java application platform is fully supported for use in both development and production, according to your corresponding subscription agreement.

Component	EAP 6.0.0	EAP 6.0.1	EAP 6.1.0	EAP 6.1.1	EAP 6.2.0	EAP 6.3.0	EAP 6.4.0
JBoss AS	7.1.2 Final	7.1.3 Final	7.2.0 Final	7.2.1 Final	7.3.0 Final	7.4.0 Final	7.5.0 Final
JBoss Web	7.0.16.Final	7.0.17.Final	7.2.0.Final	7.2.2.Final	7.2.2.Final	7.4.1.Final	7.5.0.Final

每一个EAP的版本实际上对应的是Jboss AS的版本，而对于Jboss AS中国的模块版本，也可以对应上。

对于Jboss EAP 6之后，再2016年5月份刚刚发布，其实际集成的内容已经转为了Wildfly了，对于原来的Jboss AS 系列不再支持了；

下载《开发者天堂》

下载 (/download/dev.apk)





### c.Jboss Web Server , JWS系列

在前面一节分析过的，Jboss web其实分为两种模式，一种是可以直接运行的，相当于在tomcat中加了jboss的很多优化特性，

而另外一种，就是直接集成到Jboss AS中的；

下载地址：http://www.jboss.org/products/webserver/download/

d.EWS，相当于JWS的企业版本：

## ENTERPRISE-CLASS WEB SERVER

# Create large-scale websites and lightweight web applications

Your development team loves the flexibility of open source web servers and components, like Apache and Tomcat. But as your company grows, it demands that you move to a more secure, more stable environment with the enterprise-level features you need to deliver large-scale websites and lightweight web apps.

With Red Hat, you can have it all. Red Hat® JBoss® Web Server combines the world's most deployed web server (Apache) with the top servlet engine (Tomcat) and the best support in middleware (ours).

其目标定位在，企业级的大规模访问，高并发。

..

## 总结：

Jbossweb实际包装就是Tomcat，而新版本包装的是undertow，我们在后续会以Jbossweb作为视角来详解这两个web容器。

敬请期待。

分享：

阅读 106 0

下载《开发者大全》

下载 (/download/dev.apk)



应用服务器技术讨论圈 更多文章
东方通加码大数据业务 拟募资8亿收购微智信业 (/html/308/201504/206211355/1.html)
玩转Netty – 从Netty3升级到Netty4 (/html/308/201504/206233287/1.html)
金蝶中间件2015招聘来吧！Come on！ (/html/308/201505/206307460/1.html)
GlassFish 4.1 发布，J2EE 应用服务器 (/html/308/201505/206323120/1.html)
Tomcat对keep-alive的实现逻辑 (/html/308/201505/206357679/1.html)

猜您喜欢
写给程序员的「桂林三日游」攻略 (/html/170/201609/2652974026/1.html)
微软：2016年1月以后 这些老版IE都不要了 (/html/311/201408/200580455/1.html)
数字货币到底解决了哪些问题？ (/html/343/201605/2650508755/1.html)
老师，我上课听得懂，但是敲不出来代码怎么办？ (/html/447/201505/205240021/1.html)
有些话，我只想和你说。 (/html/372/201608/2652953913/1.html)