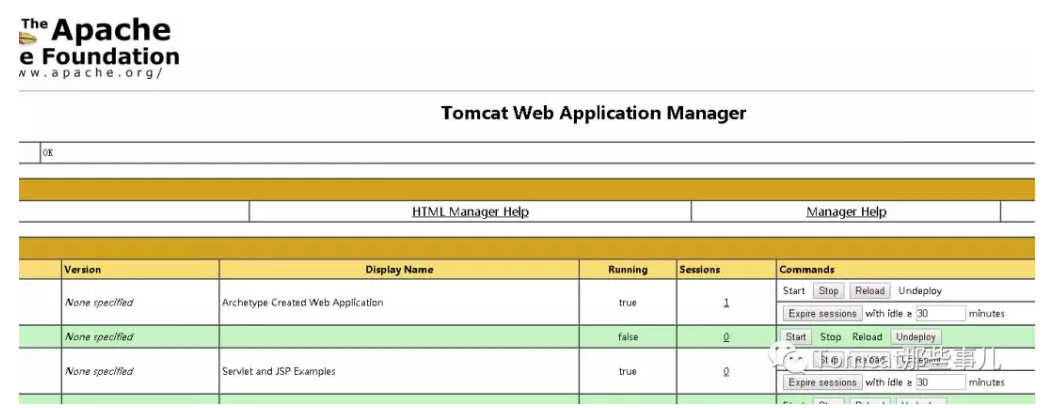
一款功能强大的Tomcat 管理监控工具

## 一款功能强大的Tomcat 管理监控工具

<https://mp.weixin.qq.com/s/AlkpgVwRD4_shLSYGmWVuw>



启动 Tomcat完毕 ，有些时候总会打开浏览器 http://localhost:8080/

去验证你的Tomcat是否启动成功，看到ROOT应用里那个黄黄的三条腿的猫，你就放心了：服务启动没问题。如果请求不成功，那一定是哪里出了问题，启动异常才导致这个默认自带的app没有启动成功。从而进一步看log分析。

除了ROOT应用外，Tomcat还默认提供了 Manager这个应用。

之前的文章里我们曾分析过 Manager 应用的实现细节

[深入Tomcat的Manager](http://mp.weixin.qq.com/s?__biz=MzI3MTEwODc5Ng==&mid=401159585&idx=1&sn=7bb6d09acfd174b9756cda4c865d2659&scene=21#wechat_redirect)

[为什么你的Manager登录不成功?](http://mp.weixin.qq.com/s?__biz=MzI3MTEwODc5Ng==&mid=2650859258&idx=1&sn=1d11be5ae19aad97ac1680dc6763c872&chksm=f1329929c645103ff95590d014973e4798bd6f9ff53c5883990bfaca5fbfdb1a1f37cf2b2031&scene=21#wechat_redirect)

在功能上 Manager 可以

1. 部署

    管理已部署的应用(启、停应用，重部署)

    部署war

    通过context.xml形式部署

    设置Session过期

2. 诊断内存泄露等

3. 显示 JVM 内存信息

4. Connector 线程的状态

还有什么呢？

我们看到， Tomcat 的这个Manager提供的，是比较常用，但很有限的几个基础功能。

使用过 比如 Weblogic, WebSphere, 开源的 GlassFish 等J2EE 应用服务器的朋友会发现，与Tomcat 的Manager 相比， 他们都提供了一个功能更完备的所谓的「**管理控制台**」。

这个管理控制台，可以做的事情，比上面 Manager 提供的就丰富多了。比如最常用的一个： 8080 这个端口不够用，我们想再加一个端口响应其他的请求，或者在配置ajp请求处理时，我们想加一个 Connector ，这个时候在 J2EE的管理控制台里，直接操作添加即可，一步步来。

甚至应用部署的时候，可以选择JSP的预编译([JSP预编译，加速你的应用](http://mp.weixin.qq.com/s?__biz=MzI3MTEwODc5Ng==&mid=2650859332&idx=1&sn=34d8a6af7ba80406401edd0b2b6bb2e9&chksm=f1329897c6451181b16647e2ee50e5d6c04947b4d3cf4bacf0f9c9e704ca6f248e0aea30cfef&scene=21#wechat_redirect))，创建数据源等等。

在 Tomcat 里对于创建 Connector 这个操作，你需要怎么做？

打开server.xml， 把 8080这个配置拷一份，改改端口，改改BIO/NIO的处理，**重启**Tomcat。(敲黑板，这里是考点哦)，而管理控制台一类的处理时不需要重启实例的。实质上Connector 做为一个组件与 Container 是不互相影响的，其实 Tomcat 也能做到不需要重启，只是这个创建 Connector 的过程，需要我们以编程的方式，来对配置文件的内容进行Marshaller/Unmarshaller。直接操作Container的内容，也可以通过JMX 来实现。

你说你部署了一个应用，该应用里配置了哪些Servlet、哪些Listener? Web.xml里有没有init-param，去哪能看看么？

不好意思，需要自己打开应用配置来看。

管理控制台的思路，一般会将当前应用具体的配置信息列出来，方便管理。

当然，区别还有很多...

其实，在Tomcat 6之前，也是有这样一款名为「Tomcat Admin」的应用，功能可以媲美应用服务器的管理控制台，不过后来不再提供了。可惜呀。

当然说了这么多，我们并不是开始嫌弃 Tomcat了。毕竟做为一个纯的Web Container，而且是非商业产品，也算不错了。

为了能使用一个较好用，功能也较全面的 Tomcat 管理工具，官方不提供我们只能把目光投向巨大的第三方开源之海，埋头苦找么？

官方的wiki里也我们指了条路：

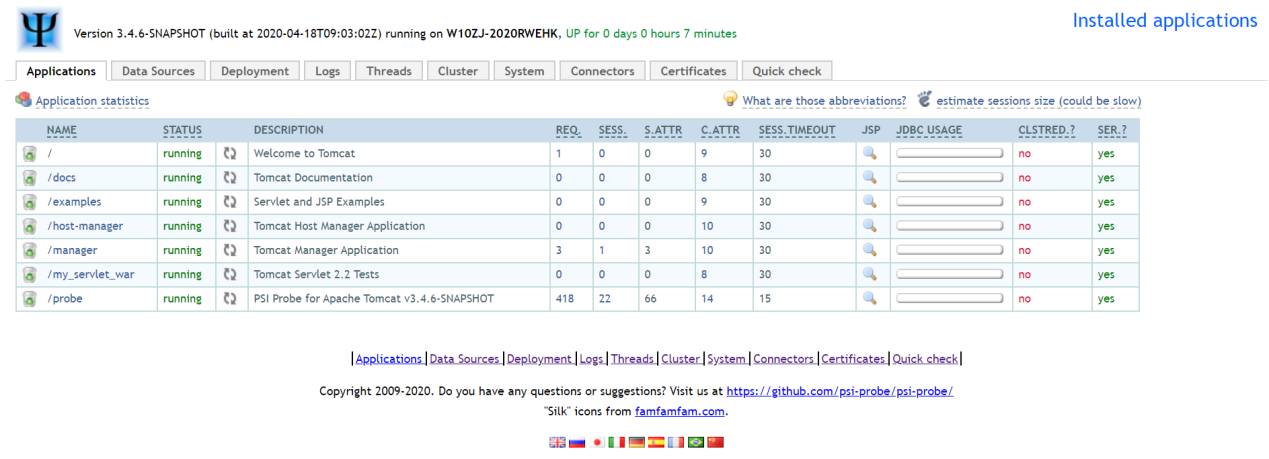
PSI Probe is **advanced manager** and **monitor** web application for Apache Tomcat.

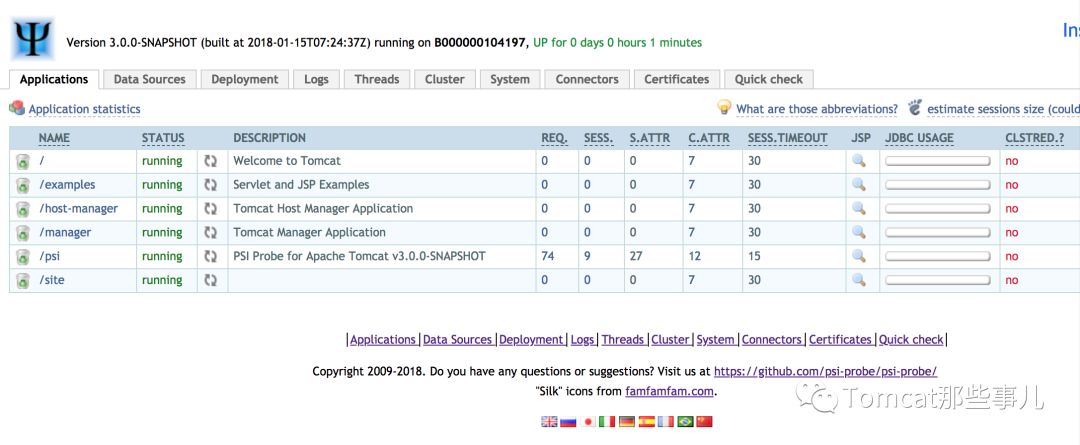
这是一款 Tomcat 管理和监控工具，前身是 Lambda Probe。由于 Lambda Probe 2006不再更新，所以 PSI Probe 算是对其的一个 Fork 版本并一直更新至今。

使用也很方便，在github(https://github.com/psi-probe/psi-probe) 下载对应的war文件，部署即可。

访问的时候，和 Manager 一样，也是需要使用配置在 **tomcat-users.xml**中的用户来访问。

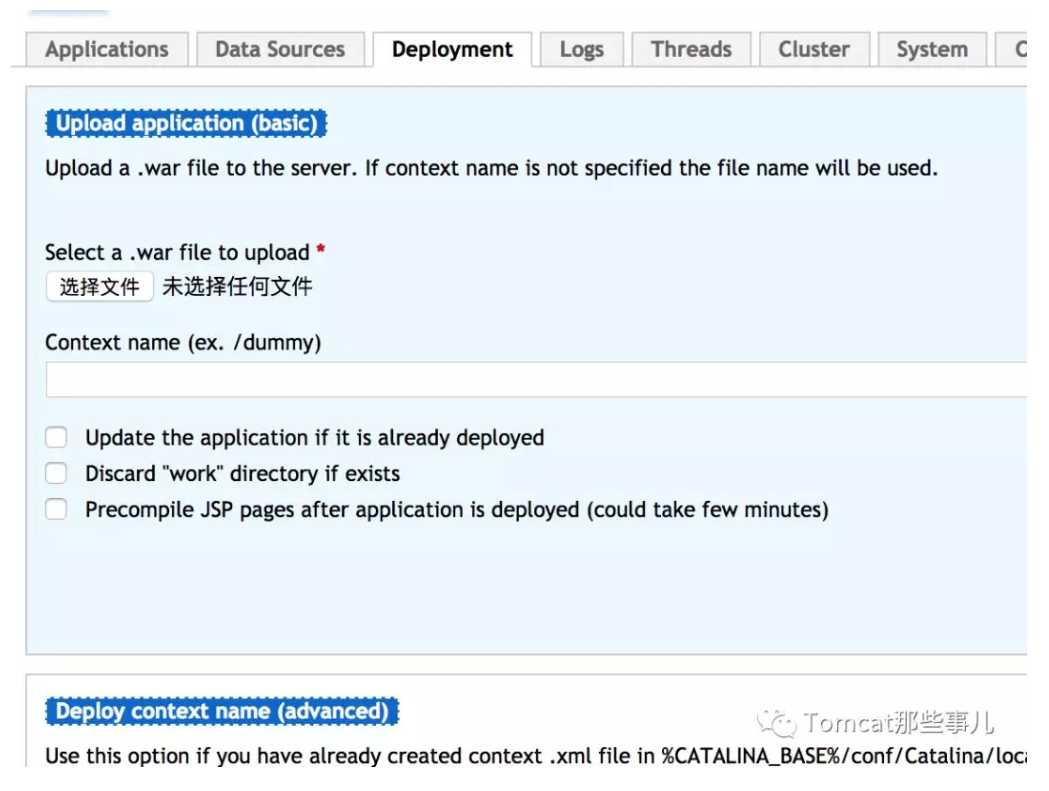
登录后是这个样子：





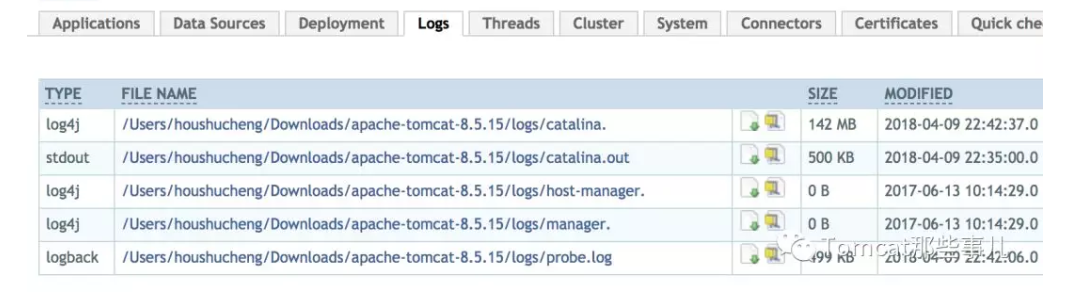
通过上面一长串的tab，就能感受到功能上要比 Manager应用要强大吧。甚至部分功能能追平「管理控制台」的体验了。

来看应用部署这一块

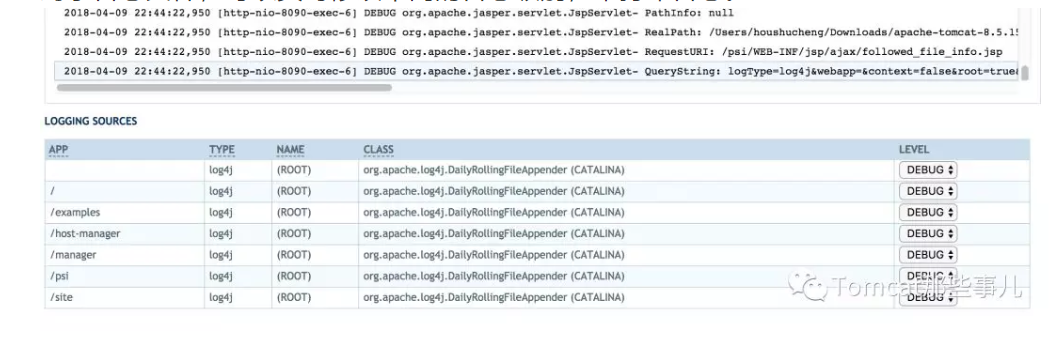


除了部署这个功能外，还包含一般J2EE的应用服务器包含的功能「**JSP预编译**」来加速应用的响应。还有清空work的选项。

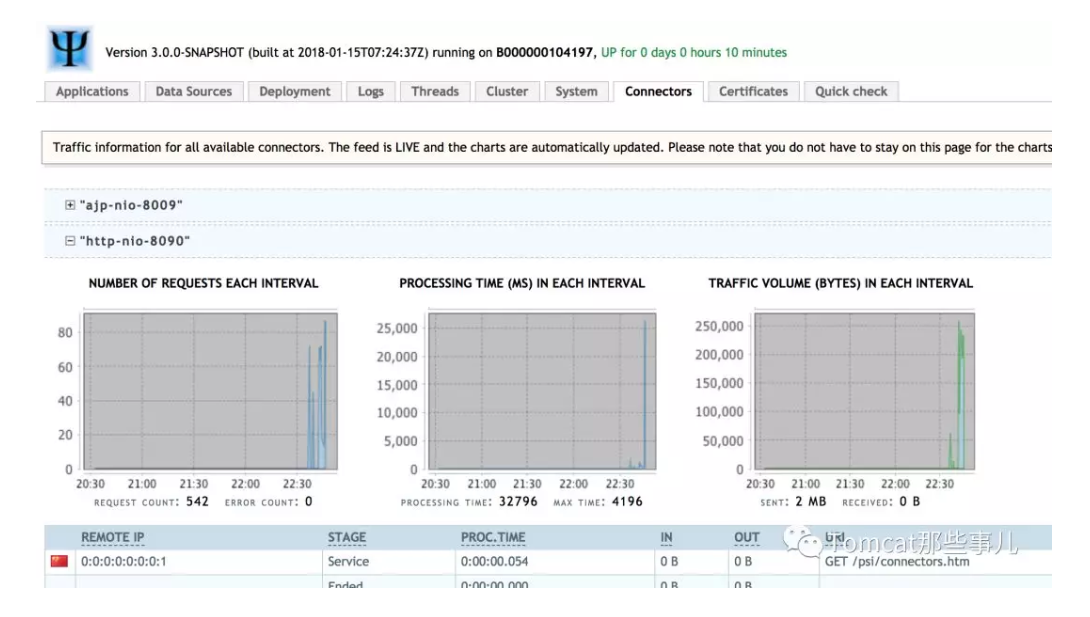
直接页面看日志



对于日志文件，可以实时修改不同的日志级别，来打印日志。



可以将当前配置的 Connector列出来，各个 Connector 收到的请求等

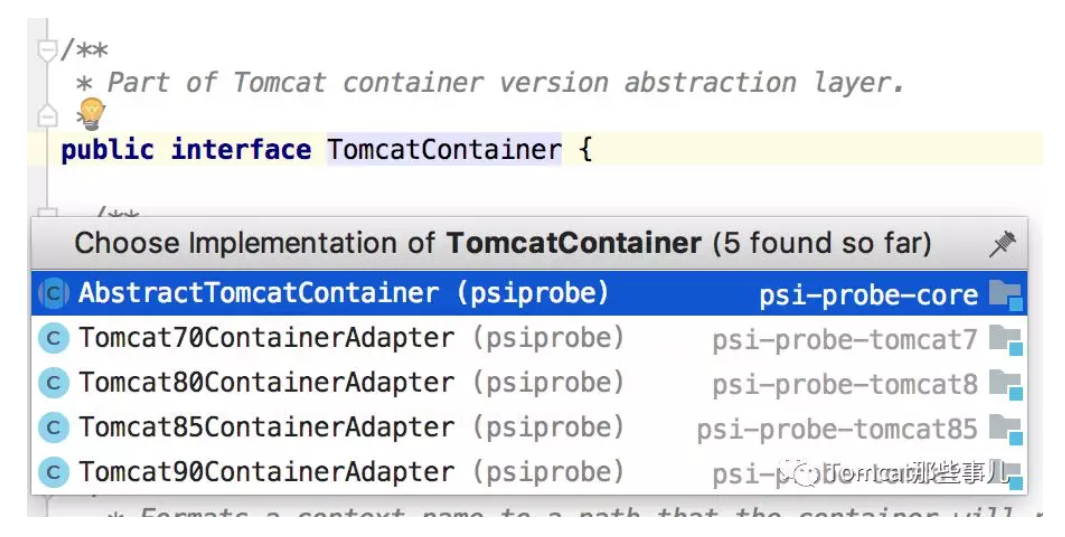


更多的功能，下载下来探索吧！ Have fun！

PSI背后是如何实现的功能呢？

我们之前的文章写过，在 Tomcat 内存在一种「**特权应用**」([啥，Tomcat里竟然还有特权应用?](http://mp.weixin.qq.com/s?__biz=MzI3MTEwODc5Ng==&mid=2650859476&idx=1&sn=8be7a37b59a5d167998f6695a1606d39&chksm=f1329807c6451111d2a1c379221655dc87dd105b067f894bfb202d1f9f283bad310a5cdc2277&scene=21#wechat_redirect))

是可以直接访问Tomcat的一些内部组件的，PSI就是把自己设置成一个特权应用，同时对于Tomcat内的操作， PSI是通过 Wrapper的形式，对不同版本的Tomcat 创建了一个Adapter



这里的 Wrapper，就是我们熟悉的Tomcat组件的那个Wrapper，PSI继承了 Spring 的DispatherServlet，通过Servlet这个Wrapper之后，再拿到其上层组件 Context、Host、Engine。

*/\*\*  
 \* Main dispatcher servlet. Spring default dispatcher servlet had to be superseded to handle  
 \* "privileged" application context features. The actual requirement is to capture passed Wrapper  
 \* instance into ContainerWrapperBean. Wrapper instance is our gateway to Tomcat.  
 \*/*@WebServlet  
**public class** ProbeServlet **extends** DispatcherServlet **implements** ContainerServlet {

在Spring 的Initializer里将Servlet注册上去

*// Set probe servlet*ServletRegistration.Dynamic probe = servletContext.addServlet(**"probe"**, ProbeServlet.**class**);

所以拿到了这些组件，在操作应用的部署，获取Connector等内容时，其实就和 Tomcat 自己一样了。

所以如果要开发管理工具，这也是一种实现思路。

一种通过JMX 的MBean，([你了解JMX在Tomcat的应用吗?](http://mp.weixin.qq.com/s?__biz=MzI3MTEwODc5Ng==&mid=401135587&idx=1&sn=610950fda2eceb3683a9fe45078f1a83&scene=21#wechat_redirect))

一种是通过PSI这种思路。