

Hello, Web Listener

作者: [liumapp](#)

原文链接: <https://ld246.com/article/1496406634488>

来源网站: [链滴](#)

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

<h2 id="Hello-Web-Listener">Hello,Web Listener</h2>

<blockquote>

<p>这是一篇供笔记性质的博文，仅用于帮助基础知识的记忆。</p>

</blockquote>

<h2 id="定义">定义</h2>

<p>web 监听器是 Servlet 规范中定义的一种特殊类，可以监听客户端的请求，也可以监听服务端的操作。监听的对象包括：ServletContent、HttpSession 以及 ServletRequest 等域对象。这三个对象分别对应 application、session 和 request 对象，可以监听它们的创建、销毁以及属性变化的事件并在这些事件发生前、发生后，做一些必要的处理。</p>

<h2 id="用途">用途</h2>

<p>统计在线人数和在线用户</p>

<p>系统启动时加载初始化信息（缓存、公用链接等）</p>

<p>统计网站访问量</p>

<p>跟 spring 结合</p>

<h2 id="创建一个web监听器">创建一个 web 监听器</h2>

<p>相关源码请见 jsp-basic</p>

<p>创建一个新的 listener 包，包下创建一个 implements 于 ServletContextListener 的类，类下现两个方法，contextInitialized 和 contextDestroyed。具体例子如：</p>

```
<pre><code class="language-go highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> <span class="highlight-kn">package</span> <span class="highlight-nx">com</span></span><span class="highlight-p">.</span><span class="highlight-nx">liumapp</span></span><span class="highlight-p">.</span><span class="highlight-nx">jspbasic</span></span><span class="highlight-p">.</span><span class="highlight-nx">listener</span></span><span class="highlight-p">;</span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="highlight-kn">import</span> <span class="highlight-nx">javax</span></span><span class="highlight-p">.</span><span class="highlight-nx">servlet</span></span><span class="highlight-p">.</span><span class="highlight-nx">ServletContextEvent</span></span><span class="highlight-p">;</span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="highlight-kn">import</span> <span class="highlight-nx">javax</span></span><span class="highlight-p">.</span><span class="highlight-nx">servlet</span></span><span class="highlight-p">.</span><span class="highlight-nx">ServletContextListener</span></span><span class="highlight-p">;</span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="highlight-cm">/**
```

```
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-cm"> * Created by liumapp on 6/1/17. * E-mail:liumapp.com@gmail.com * home-page:http://www.liumapp.com */</span><span class="highlight-nx">public</span> <span class="highlight-cm">*
```

```

class="highlight-nx">class</span> <span class="highlight-nx">FirstListener</span> <span
class="highlight-nx">implements</span> <span class="highlight-nx">ServletContextListene
</span><span class="highlight-p">{</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class="
highlight-nx">public</span> <span class="highlight-nx">void</span> <span class="highlig
t-nf">contextInitialized</span><span class="highlight-p">(</span><span class="highlight-
x">ServletContextEvent</span> <span class="highlight-nx">ServletContextEvent</span><span
class="highlight-p">)</span><span class="highlight-p">{</span>
</span></span><span class="highlight-line"><span class="highlight-cl">        <span class
"highlight-nx">System</span><span class="highlight-p">.</span><span class="highlight-n
">out</span><span class="highlight-p">.</span><span class="highlight-nb">println</spa
><span class="highlight-p">(</span><span class="highlight-s">"initialized"</span><span c
ass="highlight-p">);</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class="h
ghlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class="
highlight-nx">public</span> <span class="highlight-nx">void</span> <span class="highlig
t-nf">contextDestroyed</span><span class="highlight-p">(</span><span class="highlight
nx">ServletContextEvent</span> <span class="highlight-nx">ServletContextEvent</span><span
class="highlight-p">)</span><span class="highlight-p">{</span>
</span></span><span class="highlight-line"><span class="highlight-cl">        <span class
"highlight-nx">System</span><span class="highlight-p">.</span><span class="highlight-n
">out</span><span class="highlight-p">.</span><span class="highlight-nb">println</spa
><span class="highlight-p">(</span><span class="highlight-s">"destroyed"</span><span
lass="highlight-p">);</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class="h
ghlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class="hi
hlight-p">}</span></span>
</span></span></code></pre>

```


<p>配置 web.xml 文件，添加如下代码：</p>

```

<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl"> &lt;listener&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">    &lt;listener-cla
s&gt;com.liumapp.jspbasic.listener.FirstListenerlistener-class&gt;
</span></span><span class="highlight-line"><span class="highlight-cl"> &lt;listener&gt;
</span></span></code></pre>

```


<p>到此，第一个 web 监听器创建完成</p>

<h2 id="监听器启动顺序">监听器启动顺序</h2>

<p>多个监听器的启动顺序按照 web.xml 中的加载顺序来定义</p>

<p>优先级：监听器 > 过滤器 > Servlet</p>


```

"highlight-nx">System</span><span class="highlight-p">.</span><span class="highlight-n
">out</span><span class="highlight-p">.</span><span class="highlight-nb">println</spa
><span class="highlight-p">(</span><span class="highlight-s">"sessionCreated" </span><
pan class="highlight-p">);</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="
ighlight-cm">/**
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-cm"> * 配置文件中设置了session的超时时间为1分钟，但实际所需要的时间是1分半
会执行sessionDestroy. * @param HttpSessionEvent
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-cm"> */</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-nx">public</span> <span class="highlight-nx">void</span> <span class="highlig
t-nf">sessionDestroyed</span><span class="highlight-p">(</span><span class="highlight-
x">HttpSessionEvent</span> <span class="highlight-nx">httpSessionEvent</span><span c
ass="highlight-p">)</span> <span class="highlight-p">{</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class
"highlight-nx">System</span><span class="highlight-p">.</span><span class="highlight-n
">out</span><span class="highlight-p">.</span><span class="highlight-nb">println</spa
><span class="highlight-p">(</span><span class="highlight-s">"sessionDestroyed" </span>
<span class="highlight-p">);</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="hi
hlight-p">}</span>
</span></span></code></pre>
</li>
<li>
<p>ServletRequest-&gt;ServletRequestListener，在项目中，可以定义多个 ServletRequestListe
er，但只会会有一个 ServletRequest 对象。具体代码如下所示</p>
<pre><code class="language-go highlight-chroma"><span class="highlight-line"><span cla
s="highlight-cl"> <span class="highlight-kn">package</span> <span class="highlight-nx"
com</span><span class="highlight-p">.</span><span class="highlight-nx">liumapp</spa
><span class="highlight-p">.</span><span class="highlight-nx">jspbasic</span><span cla
s="highlight-p">.</span><span class="highlight-nx">listener</span><span class="highligh
-p">;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="hi
hlight-kn">import</span> <span class="highlight-nx">javax</span><span class="highlight
p">.</span><span class="highlight-nx">servlet</span><span class="highlight-p">.</span>
<span class="highlight-nx">ServletRequestEvent</span><span class="highlight-p">;</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="hi
hlight-kn">import</span> <span class="highlight-nx">javax</span><span class="highlight
p">.</span><span class="highlight-nx">servlet</span><span class="highlight-p">.</span>
<span class="highlight-nx">ServletRequestListener</span><span class="highlight-p">;</sp
n>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="hi
hlight-cm">/**
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla

```

```
s="highlight-cm"> * Created by liumapp on 6/2/17. * E-mail:liumapp.com@gmail.com * ho
e-page:http://www.liumapp.com */</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="hi
hlight-nx">public</span> <span class="highlight-nx">class</span> <span class="highligh
-nx">FirstServletRequestListener</span> <span class="highlight-nx">implements</span>
span class="highlight-nx">ServletRequestListener</span> <span class="highlight-p">{</sp
n>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="
ighlight-nx">public</span> <span class="highlight-nx">void</span> <span class="highlig
t-nf">requestDestroyed</span><span class="highlight-p">(</span><span class="highlight
nx">ServletRequestEvent</span> <span class="highlight-nx">servletRequestEvent</span>
span class="highlight-p">)</span><span class="highlight-p">{</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class
"highlight-nx">System</span><span class="highlight-p">.</span><span class="highlight-n
">out</span><span class="highlight-p">.</span><span class="highlight-nb">println</spa
><span class="highlight-p">(</span><span class="highlight-s">"requestDestroyed" </span
<span class="highlight-p">);</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="
ighlight-nx">public</span> <span class="highlight-nx">void</span> <span class="highlig
t-nf">requestInitialized</span><span class="highlight-p">(</span><span class="highlight-
x">ServletRequestEvent</span> <span class="highlight-nx">servletRequestEvent</span><
pan class="highlight-p">)</span><span class="highlight-p">{</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class
"highlight-nx">String</span> <span class="highlight-nx">name</span> <span class="highl
ght-p">=</span> <span class="highlight-nx">servletRequestEvent</span><span class="hi
hlight-p">.</span><span class="highlight-nf">getServletRequest</span><span class="highl
ght-p">().</span><span class="highlight-nf">getParameter</span><span class="highlight-
">(</span><span class="highlight-s">"name" </span><span class="highlight-p">);</span>
<span class="highlight-c1">// by get method
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-c1"></span> <span class="highlight-nx">System</span><span class="highl
ght-p">.</span><span class="highlight-nx">out</span><span class="highlight-p">.</spa
><span class="highlight-nb">println</span><span class="highlight-p">(</span><span clas
="highlight-s">"requestInitialized and name is : "</span><span class="highlight-o">+</spa
> <span class="highlight-nx">name</span><span class="highlight-p">);</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
ghlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="hi
hlight-p">}</span>
</span></span></code></pre>
</li>
</ul>
<h4 id="监听域对象中的属性的增加和删除的事件监听器">监听域对象中的属性的增加和删除的事
监听器</h4>
<ul>
<li>
<p>ServletContext-&gt;ServletContextAttributeListener</p>
</li>
<li>
<p>HttpSession-&gt;HttpSessionAttributeListener</p>
</li>

```



```
</li>
<p>ServletRequest-&gt;ServletRequestAttributeListener</p>
</li>
</ul>
```

上面三个监听器的用法都是一样的，代码的例子就举一个吧，下面的例子表示，监听器会监听属
的创建、删除和替换这三个事件，然后触发事件。

```
<pre> <code class="language-go highlight-chroma"> <span class="highlight-line"> <span cla
s="highlight-cl"> <span class="highlight-kn">package</span> <span class="highlight-nx">
om</span> <span class="highlight-p">.</span> <span class="highlight-nx">liumapp</span>
<span class="highlight-p">.</span> <span class="highlight-nx">jspbasic</span> <span clas
s="highlight-p">.</span> <span class="highlight-nx">listener</span> <span class="highlight
p">;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
light-kn">import</span> <span class="highlight-nx">javax</span> <span class="highlight-
">.</span> <span class="highlight-nx">servlet</span> <span class="highlight-p">.</span>
span class="highlight-nx">ServletContextAttributeEvent</span> <span class="highlight-p">;
/span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
light-kn">import</span> <span class="highlight-nx">javax</span> <span class="highlight-
">.</span> <span class="highlight-nx">servlet</span> <span class="highlight-p">.</span>
span class="highlight-nx">ServletContextAttributeListener</span> <span class="highlight-p
">;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
light-cm">/**
</span> </span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span cla
s="highlight-cm"> * Created by liumapp on 6/2/17. * E-mail:liumapp.com@gmail.com * hom
-page:http://www.liumapp.com */</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
light-nx">public</span> <span class="highlight-nx">class</span> <span class="highlight
nx">FirstServletContextAttributeListener</span> <span class="highlight-nx">implements</
pan> <span class="highlight-nx">ServletContextAttributeListener</span> <span class="high
light-p">{</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="h
ghlight-nx">public</span> <span class="highlight-nx">void</span> <span class="highlig
t-nf">attributeAdded</span> <span class="highlight-p">(</span> <span class="highlight-nx"
>ServletContextAttributeEvent</span> <span class="highlight-nx">servletContextAttribute
vent</span> <span class="highlight-p">)</span> <span class="highlight-p">{</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class=
highlight-nx">System</span> <span class="highlight-p">.</span> <span class="highlight-nx"
>out</span> <span class="highlight-p">.</span> <span class="highlight-nb">println</spa
> <span class="highlight-p">(</span> <span class="highlight-s">"servletContextAttribute ad
ed , name is : "</span> <span class="highlight-o">+</span> <span class="highlight-nx">se
vletContextAttributeEvent</span> <span class="highlight-p">.</span> <span class="highligh
-nf">getName</span> <span class="highlight-p">());</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="hi
hlight-p">}</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="h
ghlight-nx">public</span> <span class="highlight-nx">void</span> <span class="highlig
t-nf">attributeRemoved</span> <span class="highlight-p">(</span> <span class="highlight
nx">ServletContextAttributeEvent</span> <span class="highlight-nx">servletContextAttribu
eEvent</span> <span class="highlight-p">)</span> <span class="highlight-p">{</span>
```

```

</span></span><span class="highlight-line"><span class="highlight-cl">    <span class="highlight-nx">System</span><span class="highlight-p">.</span><span class="highlight-nx">out</span><span class="highlight-p">.</span><span class="highlight-nb">println</span></span><span class="highlight-p">(</span><span class="highlight-s">"servletContextAttribute removed , name is : "</span><span class="highlight-o">+</span><span class="highlight-nx">servletContextAttributeEvent</span><span class="highlight-p">.</span><span class="highlight-p">.</span><span class="highlight-nf">getName</span><span class="highlight-p">());</span></span></span></span><span class="highlight-line"><span class="highlight-cl">    <span class="highlight-p">}</span></span></span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class="highlight-nx">public</span>    <span class="highlight-nx">void</span>    <span class="highlight-nf">attributeReplaced</span><span class="highlight-p">(</span><span class="highlight-nx">ServletContextAttributeEvent</span>    <span class="highlight-nx">servletContextAttributeEvent</span><span class="highlight-p">)</span><span class="highlight-p">{</span></span></span></span><span class="highlight-line"><span class="highlight-cl">    <span class="highlight-nx">System</span><span class="highlight-p">.</span><span class="highlight-nx">out</span><span class="highlight-p">.</span><span class="highlight-nb">println</span></span><span class="highlight-p">(</span><span class="highlight-s">"servletContextAttribute replaced , name is : "</span><span class="highlight-o">+</span><span class="highlight-nx">servletContextAttributeEvent</span><span class="highlight-p">.</span><span class="highlight-p">.</span><span class="highlight-nf">getName</span><span class="highlight-p">());</span></span></span></span><span class="highlight-line"><span class="highlight-cl">    <span class="highlight-p">}</span></span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-p">}</span></span></span></span></pre>

```

监听绑定到HttpSession域中的某个对象的的状态的事件监听器

HttpSession 中的对象状态分为:

-

-

- 绑定- > 解除绑定

- 绑定就是通过 session 的 setAttribute 添加一个 session 属性, 解除绑定就是通过 removeAttribute 删除一个 session 属性。

-

-

- 钝化- > 活化

- 钝化就是将 session 持久性的保存在 database 或者 redis 上面, 活化就是将 session 从 database 或者 redis 上面恢复到原有的保存方式。

-

-

session钝化机制

把服务器中不经常使用的 session 对象暂时序列化到文件系统或是数据库系统中, 当被使用时反序列化到内存中, 整个过程由服务器自动完成。

Tomcat中的两种session钝化管理机制

-

-

- org.apache.catalina.session.StandardManager

-

- 当 Tomcat 服务器被关闭或者重启时, Tomcat 服务器会将当前内存中的 Session 对象钝化到服务器文件系统中;

- 当 Web 应用程序被重新加载时, 内存中的 Session 对象也会被钝化到服务器的文件系统中;

- 钝化后的文件被保存在 Tomcat 安装路径/work/Catalina/hostname/applicationname/SESSI

N.ser

<p>org.apache.catalina.session.PersistentManager</p>

首先在钝化的基础上进行了扩张。第一种情况如上面的第一点，第二种情况如上面的第二点，第三种情况，可以配置主流内存的 Session 对象数目，将不常使用的 Session 对象保存到文件系统或者数据库，当用时再加载;

默认情况下，Tomcat 提供两个钝化驱动类：org.apache.catalina.FileStore 和 org.apache.catalina.JdbcStore。

<h2 id="好像扯远了-----">好像扯远了.....</h2>

<h2 id="现在回到监听器上来-----">现在回到监听器上来.....</h2>

<p>请注意，以下两个监听器不需要在 web.xml 进行注册。</p>

<p>HttpSessionBindingListener</p>

绑定:valueBound 方法

解除绑定:valueUnbound 方法

<p>实现这个接口的时候，并不是创建一个 Listener 来实现，而是创建一个普通的 JavaBean 来实现。因为最终修改的是一个普通的对象的状态，而不是监听器的状态。这里我们举的例子如下：</p>

```
<pre><code class="language-go highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> <span class="highlight-kn">package</span> <span class="highlight-nx">com</span><span class="highlight-p">.</span><span class="highlight-nx">liumapp</span><span class="highlight-p">.</span><span class="highlight-nx">jspbasic</span><span class="highlight-p">.</span><span class="highlight-nx">entity</span><span class="highlight-p">.</span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="highlight-kn">import</span> <span class="highlight-nx">javax</span><span class="highlight-p">.</span><span class="highlight-nx">servlet</span><span class="highlight-p">.</span><span class="highlight-nx">http</span><span class="highlight-p">.</span><span class="highlight-nx">HttpSessionBindingEvent</span><span class="highlight-p">.</span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="highlight-kn">import</span> <span class="highlight-nx">javax</span><span class="highlight-p">.</span><span class="highlight-nx">servlet</span><span class="highlight-p">.</span><span class="highlight-nx">http</span><span class="highlight-p">.</span><span class="highlight-nx">HttpSessionBindingListener</span><span class="highlight-p">.</span></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="highlight-cm">/**
```

```
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-cm"> * Created by liumapp on 6/2/17. * E-mail:liumapp.com@gmail.com * homepage:http://www.liumapp.com */</span>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="highlight-nx">public</span> <span class="highlight-nx">class</span> <span class="highlight-nx">User</span> <span class="highlight-nx">implements</span> <span class="highlight-nx">HttpSessionBindingListener</span> <span class="highlight-p">{</span>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```



```

highlight-p">;</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
highlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="
ighlight-nx">public</span> <span class="highlight-nx">void</span> <span class="highlig
t-nf">setUserName</span><span class="highlight-p">(</span><span class="highlight-nx"
String</span> <span class="highlight-nx">userName</span><span class="highlight-p">)<
span> <span class="highlight-p">{</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class
"highlight-nx">this</span><span class="highlight-p">.</span><span class="highlight-nx"
userName</span> <span class="highlight-p">=</span> <span class="highlight-nx">userN
me</span><span class="highlight-p">;</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
highlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="
ighlight-nx">public</span> <span class="highlight-nx">void</span> <span class="highlig
t-nf">setPassWord</span><span class="highlight-p">(</span><span class="highlight-nx"
String</span> <span class="highlight-nx">passWord</span><span class="highlight-p">)<
span> <span class="highlight-p">{</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class
"highlight-nx">this</span><span class="highlight-p">.</span><span class="highlight-nx"
passWord</span> <span class="highlight-p">=</span> <span class="highlight-nx">passW
rd</span><span class="highlight-p">;</span>
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="h
highlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="hi
hlight-p">}</span>
</span></span></code></pre>

```

<p>随后我们通过下面的两行代码即可实现 bound 和 unbound 事件的触发: </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl"> request.getSession().setAttribute("currentUser" , new com.liumapp.jspbasic.entity.User()
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> request.getSessi
n().removeAttribute("currentUser");
</span></span></code></pre>

```


<p>HttpSessionActivationListener</p>

钝化:sessionWillPassivate 方法

活化:sessionDidActive 方法

<p>代码如下所示: </p>

```
<pre><code class="language-go highlight-chroma"><span class="highlight-line"><span cla
s="highlight-cl"> <span class="highlight-kn">package</span> <span class="highlight-nx"
com</span><span class="highlight-p">.</span><span class="highlight-nx">liumapp</spa
><span class="highlight-p">.</span><span class="highlight-nx">jspbasic</span><span cla
s="highlight-p">.</span><span class="highlight-nx">entity</span><span class="highlight-
">;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
```



```
</span></span></code></pre>
```

<p>我们进入 init 页面后停止 tomcat 服务器，session 会自动序列化之后存入文件系统中，相关信息如下所示：</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> sessionWillPassivate : org.apache.catalina.session.StandardSessionFacade@4666d1cf</span></span></code></pre>
```

<p>第二次启动 Tomcat 之后，session 信息将会反序列化然后被加载，我们可以在 index.jsp 看到入的 session 信息，同时请注意，这里还需要引入 Serializable 的接口实现，不然无法成功反序列化</p>

<p>上面的描述都是 Tomcat 的第一种 session 钝化管理机制，那么第二种 PersistentManager 如何使用呢？</p>

<p>首先我们进入 Tomcat/conf 目录，打开 context.xml 文件，找到如下的代码</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">&lt;!-- Uncomment this to disable session persistence across Tomcat restarts --&gt;</span></span><span class="highlight-line"><span class="highlight-cl">&lt;!--</span></span><span class="highlight-line"><span class="highlight-cl">&lt;Manager pathame="" /&gt;</span></span><span class="highlight-line"><span class="highlight-cl">--&gt;</span></span></code></pre>
```

<p>把 manager 的 pathname 设置为 FileStore 或者 JdbcStore，两者类型的详细信息下回分解。</p>

</p>

<p>前文的描述都是针对 Servlet2.5 版本来进行的，那么在 Servlet3.0 下面，我们需要做的，就是需要在 web.xml 下面进行 listener 的任何注解，只需要在 listener 类的前面，加一个注解即可，如： </p> ``` <pre><code class="highlight-chroma">@WebListener("this is my web listener")class</code></pre> ``` 原文链接：[Hello,Web Listener](#)