**详解JavaEE使用过滤器实现登录(用户自动登录 安全登录 取消自动登录黑用户禁止登录)**

转载  2016-05-19   作者：YX\_blog   [[http://www.jb51.net/skin/2018/images/text-message.png](http://www.jb51.net/article/84587.htm#comments) 我要评论](http://www.jb51.net/article/84587.htm#comments)

主要介绍用户的自动登录和取消自动登录，以及实现一天自动登录或者n天实现自动登录，当用户ip被加入到黑名单之后，直接利用过滤器返回一个警告页面。接下来通过本文给大家介绍JavaEE使用过滤器实现登录的相关知识，感兴趣的朋友一起学习吧

在我们生活中，对于账户的自动登录已经很常见了，所以利用过滤器实现这个功能。

主要介绍用户的自动登录和取消自动登录，以及实现一天自动登录或者n天实现自动登录，当用户ip被加入到黑名单之后，直接利用过滤器返回一个警告页面。

过滤器的功能很是强大，我们只需要在写好的前台后servlet之后进行添加就可以实现这个功能

Ps:这个仅仅只是一个演示而已，里面的访问数据库的部分，自己随意模拟了下，主要是突出实现自动登录的功能。

**前台代码：**

前台代码是成功与否都在这个页面显示。用到的技术：jstl标签的应用，session读取值

[?](http://www.jb51.net/article/84587.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | <%@ page language="java" import="java.util.\*" pageEncoding="utf-8"%>  <%@taglib uri="<http://java.sun.com/jsp/jstl/core>" prefix="c"%>  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  <html>  <head>  </head>  <body>  <c:if test="${!empty sessionScope.error }">  你的密码或用户名错误。<!-- 显示后就需要把里面的值移走 -->  <c:remove var="error" scope="session"/>  </c:if>  <c:if test="${empty sessionScope.user }" var="boo">  <h2>这是登录的页面</h2>  <form action="<c:url value='/LoginServlet'/>" method="post">  NAME:<input type="text" name="name" /><br/>  PWD:<input type="text" name="pwd" /><br/>  不自动登录:<input type="radio" name="time" value="0" /><br/>  一天:<input type="radio" name="time" value="1" /><br/>  七天:<input type="radio" name="time" value="7" /><br/>  <input type="submit" value="提交" />  </form>  </c:if>  <c:if test="${!boo }">  欢迎您，${sessionScope.user },登录成功  <a href="">模块一 </a>  <a href="">模块2 </a>  <a href="<c:url value='/CancelAutoLogin'/>">取消自动登录</a>  </c:if>  </body>  </html> |



**servlet的实现代码：**

和以前的代码一样，只负责和前台交互即可：里面用到的技术有url编码，值存在cookie里面，存在session里面，页面跳转（转发）

[?](http://www.jb51.net/article/84587.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | public void doPost(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  String name=request.getParameter("name");  String pwd=request.getParameter("pwd");  String time=request.getParameter("time");  if(name!=null && pwd!=null && name.equals(pwd)){//此处随意写写，后面应该去servvice-->dao访问数据库  //这里假设登录成功了，我们把信息存入session里面  request.getSession().setAttribute("user", name);  //兼容中文，我们需要进行编码  name=URLEncoder.encode(name, "utf-8");  pwd=URLEncoder.encode(pwd, "utf-8");  Cookie c =new Cookie("autologin", name+","+pwd);//这个value不能采用这种方式的，安全性考虑，我们必须知道采用加密，或者二次加密，  int \_time=60\*60\*24\*Integer.valueOf(time);  c.setMaxAge(\_time);  response.addCookie(c);  response.sendRedirect(request.getContextPath()+"/index.jsp");//在过滤器中默认的设置是拦截重定向，转发是内部直接转发，不过过滤器，不好办，但是只需要在web.xml中配置就可以了。  }else{  request.getSession().setAttribute("error", "1");  response.sendRedirect(request.getContextPath()+"/index.jsp");  }  } |

到现在为止，都感觉没有什么技术，和以前的代码一个，现在就是Filter的作用了。

**安全登录：**

之前我们采用过动态导入来进行安全登录，防止用户进入项目之后，不用登录，随意输入都可以进入界面，动态导入能够实现这个功能，但是，采用过滤器更好。

在过滤器中一般写的都是dofilter();只需要判断session容器里面是否为null,为null这说明这是没有登录的，直接踢回登录界面，否，则放行

**代码呈上：**

[?](http://www.jb51.net/article/84587.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | public void doFilter(ServletRequest request, ServletResponse response,  FilterChain chain) throws IOException, ServletException {  HttpServletRequest req=(HttpServletRequest) request;  HttpServletResponse resp=(HttpServletResponse) response;  String session=(String) req.getSession().getAttribute("user");  if(session==null){  System.out.println("非正常登录");  resp.sendRedirect(req.getContextPath()+"/index.jsp");  }else{  System.out.println("成功登录");  chain.doFilter(req, resp);  }  } |

**字符编码：**

字符编码的问题，以前每一次都需要在servlet的dopost（）里面自己手动输入，request.setCharacterEncoding("utf-8");每一个servlet都需要输入，太麻烦，我们采用过滤器实现；

**代码呈上：**

[?](http://www.jb51.net/article/84587.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | <span style="font-size:18px;">public void doFilter(ServletRequest request, ServletResponse response,  FilterChain chain) throws IOException, ServletException {  request.setCharacterEncoding(character);//去客户端接收的编码  response.setContentType("text/html;charset=utf-8");//设置发出去的编码  chain.doFilter(request, response);  }    @Override  public void init(FilterConfig config) throws ServletException {  character=config.getInitParameter("character");//a</span><span style="font-size: 18px; font-family: Arial, Helvetica, sans-serif;">haracter 设置为全局变量，</span><span style="font-size:18px;">  }</span> |

再上面的character定义为全局变量，初始值在web.xml中配置。

**web.xml代码呈上：**

[?](http://www.jb51.net/article/84587.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <filter>  <filter-name>character</filter-name>  <filter-class>cn.hncu.Filter.CharacterFilter</filter-class>  <init-param>  <param-name>character</param-name>  <param-value>UTF-8</param-value>  </init-param>  </filter> |

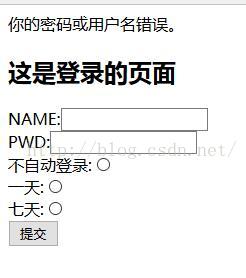
**自动登录：**

主要思路：自动登录需要判断session里面是都存有值，有，则登录过了，没有，就去本地cookie查找，存在，去数据库匹配，若匹配成功，就把session容器添加值。

**代码呈上：**

[?](http://www.jb51.net/article/84587.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | public void doFilter(ServletRequest request, ServletResponse response,  FilterChain chain) throws IOException, ServletException {  //自动登录，必须要设置session里面是都有值，有，则当前登录过，没有，就要去访问cookie里面的数据，cookie里面的数据  //是否和数据库里面的匹配，是，将session里面的值在这里设置，否，放走  HttpServletRequest req=(HttpServletRequest) request;  HttpServletResponse resp =(HttpServletResponse) response;  String session =(String) req.getSession().getAttribute("user");  if(session==null){//说明当前没有登录过  Cookie cs[]=req.getCookies();  if(cs!=null){  for(Cookie c:cs){  if(c.getName().equals("autologin")){  String value=c.getValue();//这是经过加密的，但是我们仅仅只是采用逗号连接了一下。  String[] strs=value.split(",");//在logserlvet里面采用的是先编码，再采用逗号连接，我们这里需要反过来  String name=URLDecoder.decode(strs[0], "utf-8");  String pwd=URLDecoder.decode(strs[1], "utf-8");  //将name,pwd数据拿到后台访问数据库，我们这里只是随便写写  if(name.equals(pwd)){  req.getSession().setAttribute("user", name);//设置session里面的参数  break;  }  }  }  }  }  chain.doFilter(req, resp);//一定要放走哦。。  } |



**黑名单用户**

黑名单用户，不准登录，直接告诉它结果

**代码呈上：**

[?](http://www.jb51.net/article/84587.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | public void doFilter(ServletRequest request, ServletResponse response,  FilterChain chain) throws IOException, ServletException {  HttpServletRequest req=(HttpServletRequest) request;  HttpServletResponse resp=(HttpServletResponse) response;  String ip=req.getRemoteAddr();//获取访问的ip;  System.out.println(ip+"IIPP");  if(set.contains(ip)){//在黑名单之内  System.out.println("set");  resp.getWriter().print("您属于黑名单..<a href='"+req.getContextPath()+"/index.jsp'>返回</a>");  //返回也是不行的，因为index向服务器请求的时候就直接拦截了  }else{  chain.doFilter(req, resp);  }  } |

黑名单返回的类型为list最好，我这里是自己手动添加，原本应该从写一个工具类从数据库读取，不止能查，还能增删改--黑名单。

**代码呈上：**

Hashset 定义为全局变量，set里面含有contain，效率很高。

[?](http://www.jb51.net/article/84587.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | public void init(FilterConfig arg0) throws ServletException {  //这里是黑名单列表，从数据库中调取出来。这里只是简单的模拟下  set.add("192.132.0.12");//这是黑IP,这个是从后台数据库拿到的。  set.add("localhost");  set.add("192.132.32.4");  set.add("127.0.0.1");  } |



**取消自动登录**

当一直自动登录认为不安全，所以我们设置没有自动登录

之前我们知道，自动登录靠的是cookie里面存的技术，所以这里我们只需要将cookie删除就可以了

因为取消自动登录是超链接，所以写的是servlet.

**代码呈上：**

[?](http://www.jb51.net/article/84587.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | public void doPost(HttpServletRequest req, HttpServletResponse resp)  throws ServletException, IOException {  Cookie cc=new Cookie("autologin", "");//删除cookie的方法，就建立一个同名connkie，然后设置cookie的setmaxage=0；  cc.setMaxAge(0);  cc.setPath(req.getContextPath());  resp.addCookie(cc);  resp.sendRedirect(req.getContextPath()+"/index.jsp");  } |

上面这些就能实现这些简答的功能了。

以上所述是小编给大家介绍的详解JavaEE使用过滤器实现用户自动登录 安全登录 取消自动登录黑用户禁止登录的相关知识，希望对大家有所帮助，如果大家有任何疑问请给我留言，小编会及时回复大家的。在此也非常感谢大家对脚本之家网站的支持！

**能使JSP页面延迟的语句：**

**Thread.currentThread().sleep(180000);**

jsp页面可以利用href调用javascript方法进行延时来实现点击连接后延迟跳转。  
比如在页面上可以这么写：  
<a href="javascript:xxx();">下一页</a>  
//设置延时函数  
function sleep(numberMillis) {  
var now = new Date();  
var exitTime = now.getTime() + numberMillis;  
while (true) {  
now = new Date();  
if (now.getTime() > exitTime)  
return;  
}  
然后调用sleep(1000)xxx();

## 其他回答

$(document).ready(function(){  
setTimeout(function(){  
//此处添加a标签是否[跳转代码](https://www.baidu.com/s?wd=%E8%B7%B3%E8%BD%AC%E4%BB%A3%E7%A0%81&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dWPWmLmyf1uhnzP1f3nH6z0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHbsPWnsrHm1" \t "_blank)  
// 由于a标签没有可以可用不可用，可以添加属性来 判断，如果满足条件 href="你的地址"  
  
// 不满足就直接href="#" (你默认值可设置为#)  
  
clearTimeout(this);  
  
},1000);  
  
});