

## springboot整合shiro-配置记住我(四)

原文地址，转载请注明出处：[https://blog.csdn.net/qq\\_34021712/article/details/80306442](https://blog.csdn.net/qq_34021712/article/details/80306442) ©王赛超

记住我功能在各网站是比较常见的，实现起来也都差不多，主要就是利用cookie来实现，而shiro对记住我功能的实现也是比较简单的，只需要几步即可。Shiro提供了记住我（RememberMe）的功能，比如访问一些网站时，关闭了浏览器下次再打开时还是能记住你是谁，下次访问时无需再登录即可访问，基本流程如下：

- （1）、首先在登录页面选中RememberMe然后登录成功；如果是浏览器登录，一般会把RememberMe的Cookie写到客户端并保存下来；
- （2）、关闭浏览器再重新打开；会发现浏览器还是记住你的；
- （3）、访问一般的网页服务器端还是知道你是谁，且能正常访问；

### Shiro配置记住我步骤：

#### 在ShiroConfig中添加以下Bean

##### 记住我cookie

```
1  /**
2   * cookie对象;会话Cookie模板 ,默认为： JSESSIONID 问题：与SERVLET容器名冲突,重新定义为sid或rememberMe, 自定义
3   * @return
4   */
5  @Bean
6  public SimpleCookie rememberMeCookie(){
7      //这个参数是cookie的名称，对应前端的checkbox的name = rememberMe
8      SimpleCookie simpleCookie = new SimpleCookie("rememberMe");
9      //setcookie的httponly属性如果设为true的话，会增加对xss防护的安全系数。它有以下特点：
10
11      //setcookie()的第七个参数
12      //设为true后，只能通过http访问， javascript无法访问
13      //防止xss读取cookie
14      simpleCookie.setHttpOnly(true);
15      simpleCookie.setPath("/");
16      //<!-- 记住我cookie生效时间30天 ,单位秒;-->
17      simpleCookie.setMaxAge(2592000);
18      return simpleCookie;
19  }
```

##### 记住我管理器

```
1  /**
2   * cookie管理对象;记住我功能, rememberMe管理器
3   * @return
4   */
5  @Bean
6  public CookieRememberMeManager rememberMeManager(){
7      CookieRememberMeManager cookieRememberMeManager = new CookieRememberMeManager();
8      cookieRememberMeManager.setCookie(rememberMeCookie());
9      //rememberMe cookie加密的密钥 建议每个项目都不一样 默认AES算法 密钥长度(128 256 512 位)
10     cookieRememberMeManager.setCipherKey(Base64.decode("4AvVhmFLUs0KTA3Kprsdag=="));
11     return cookieRememberMeManager;
12  }
```

##### 记住我Filter

```
1  /**
2   * FormAuthenticationFilter 过滤器 过滤记住我
3   * @return
4   */
5  @Bean
6  public FormAuthenticationFilter formAuthenticationFilter(){
7      FormAuthenticationFilter formAuthenticationFilter = new FormAuthenticationFilter();
8      //对应前端的checkbox的name = rememberMe
9      formAuthenticationFilter.setRememberMeParam("rememberMe");
10     return formAuthenticationFilter;
11  }
```

#### 在SecurityManager中打开记住我,之前是注释掉的

```
1  /**
2   * 配置核心安全事务管理器
3   * @param shiroRealm
4   * @return
5   */
6  @Bean(name="securityManager")
7  public SecurityManager securityManager(@Qualifier("shiroRealm") ShiroRealm shiroRealm) {
8      DefaultWebSecurityManager securityManager = new DefaultWebSecurityManager();
```

```

9      //设置自定义realm.
10     securityManager.setRealm(shiroRealm);
11     //配置记住我 参考博客:
12     securityManager.setRememberMeManager(rememberMeManager());
13
14     //配置 redis缓存管理器 参考博客:
15     //securityManager.setCacheManager(getEhCacheManager());
16
17     //配置自定义session管理, 使用redis 参考博客:
18     //securityManager.setSessionManager(sessionManager());
19
20     return securityManager;
21 }

```

修改shirFilter中拦截请求的规则,将/\*\*从authc 改为user

```

1  @Bean(name = "shirFilter")
2  public ShiroFilterFactoryBean shiroFilter(@Qualifier("securityManager") SecurityManager securityManager) {
3
4      .....
5
6      // 配置访问权限 必须是LinkedHashMap, 因为它必须保证有序
7      // 过滤链定义, 从上向下顺序执行, 一般将 /**放在最为下边 --> : 这是一个坑, 一不小心代码就不好使了
8      LinkedHashMap<String, String> filterChainDefinitionMap = new LinkedHashMap<>();
9
10     //其他资源都需要认证 authc 表示需要认证才能进行访问 user表示配置记住我或认证通过可以访问的地址
11     filterChainDefinitionMap.put("/**", "user");
12
13     .....
14
15
16     return shiroFilterFactoryBean;
17 }

```

更改LoginController中login方法

接受前台记住我参数,然后传给usernamePasswordToken, 将之前的那行注释掉。

```

1  @RequestMapping(value = "/login",method = RequestMethod.POST)
2  public String loginUser(HttpServletRequest request, String username, String password,boolean rememberMe, Model model, HttpSession sess
3
4      //对密码进行加密
5      //password=new SimpleHash("md5", password, ByteSource.Util.bytes(username.toLowerCase() + "shiro"),2).toHex();
6      //如果有点击 记住我
7      UsernamePasswordToken usernamePasswordToken=new UsernamePasswordToken(username,password,rememberMe);
8      //UsernamePasswordToken usernamePasswordToken = new UsernamePasswordToken(username,password);
9      Subject subject = SecurityUtils.getSubject();
10     try {
11         //登录操作
12         subject.login(usernamePasswordToken);
13         User user=(User) subject.getPrincipal();
14         //更新用户登录时间, 也可以在ShiroRealm里面做
15         session.setAttribute("user", user);
16         model.addAttribute("user",user);
17         return "index";
18     } catch(Exception e) {
19         //登录失败从request中获取shiro处理的异常信息 shiroLoginFailure:就是shiro异常类的全类名
20         String exception = (String) request.getAttribute("shiroLoginFailure");
21         model.addAttribute("msg",e.getMessage());
22         //返回登录页面
23         return "login";
24     }
25 }

```

UserInfo实体类实现序列化

UserInfo实体类必须实现序列化接口,否则报序列化异常。因为RememberMe会将用户信息加密然后以Cookie保存。

前台页面添加记住我复选框

```

1  <!DOCTYPE html>
2  <html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org"
3      xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3"
4      xmlns:shiro="http://www.pollix.at/thymeleaf/shiro">
5  <head>
6      <meta charset="UTF-8" />
7      <title>Insert title here</title>

```

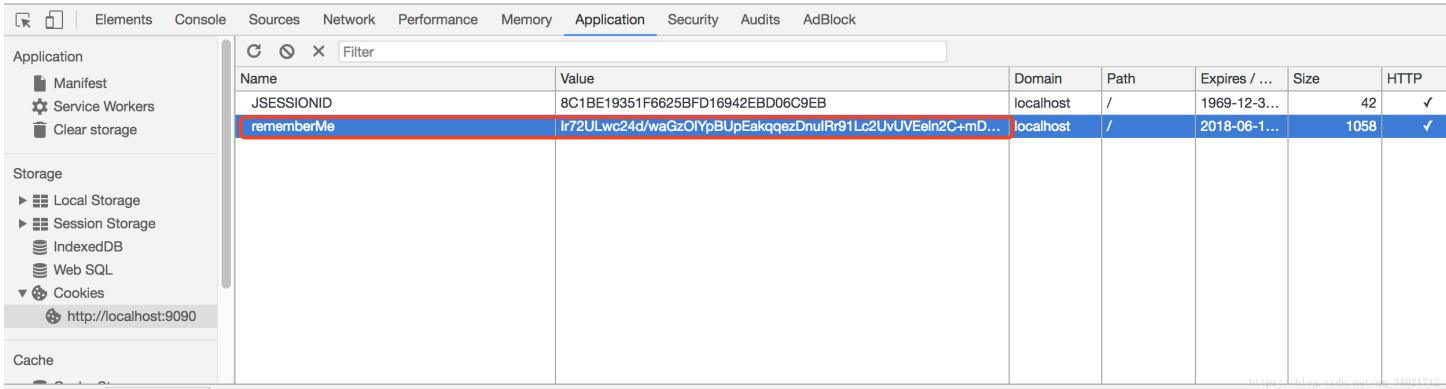
```
8 </head>
9 <body>
10 <h1>欢迎登录</h1>
11 <h1 th:if="${msg != null}" th:text="${msg}" style="color: red"></h1>
12 <form action="/login" method="post">
13     用户名: <input type="text" name="username"/><br/>
14     密码: <input type="password" name="password"/><br/>
15     <input type="checkbox" name="rememberMe" />记住我<br/>
16     <input type="submit" value="提交"/>
17 </form>
18 </body>
19 </html>
```

进行测试

上面的过程已经配置完成,现在进行测试, 重启项目,使用admin登录,然后访问: <http://localhost:9090/userInfo/view>会跳转到登录页面,登录之后退出浏览器,再次点开访问刚才的页面,可以访问。

浏览器cookie查看

重新访问可以访问,我们可以使用浏览器自带的调试功能查看浏览器保存的cookie是什么样的,如下:



rememberMe 大概内容如下:

```
1 Ir72ULwc24d/waGzOIYpBUPEakqgezDnuIRr91Lc2UvUVEeln2C+mDICIE338+vZiycabYe0KWuAY7x4PukL09KeiUu15sH0PJ04bkpYVd4HcoITCi54+KqQhL/47l007ucM0B
2 itW1IfhqzCJZjUH5nU7J+kG7NQ40f1f2HpZHH0Mlg5zTjCT0JK8Zib2TRd74rYWNnFxtz/
3 u2C1BrSYGvwdNcI0oouY3CjpJ18PmI6W4MZC4LwzXBPi90Ev8AchhVRVaCvPrGL52goZr1i0XcM2e0aJoVe5FyBnaD4EWjwFYIZMUPEGXHCsbWsyKp00mQT+/7rQRxd3rq/
4 v6r+H291CYsq1NfdRDetozMjSVtAMPvceCPbH8tfeHTLbb7B/g1xrsW0BxvsSAWdZm9t6yWP4NREe9bxaPmbmRMklssHLwG63tBgBS2/S0vAcEer0cLJmYIub+3LhVfRwvgjQp
5 0mcSZ7t4ZY8HB+L7ataFZC3aL6s49gbHUhood8yFRreS1yrEqqiucecd+vZFPn4GxHAYayeyI7VXvoaUa8AU6WQ16TBLDXvft0cQQ7maBetfitl7CtZ5pBHDU5YxF6sM5mSH+h
6 0jNZu4IZfpJrfdk+NvgHaSyfvVbo8eiedgScATiJ2nzP6NVydWs370LVRazN7LuGcUAz8pd0rLCtwshY/2/jFQNRsUpzuB3Ml1A0rFCVnhyz7bvXqS3+lzQsTshN370sf35D0d
7 7K1M33gEdpUL1tkda6435EsaCfywRKDy2604LkcWTLAmDLhvE0GXkxp3guQaZb8VTMTQA7XNU2pzlAdxPJ4JelAyxExF0S956eGZKbB9qum2+R0c++nb34oAL/9yo0b9eHgh0y
```

其实是将用户信息加密后放到前台的(包含密码), 在项目中user对象信息过于庞大, 不能全部存入Cookie, Cookie对长度有一定的限制。