



链滴

# 记录一次 Selenium WebDriver 无法正常加载 www.xxx.com 域名 Cookie 文件的解决办法

作者: [zxniuniu](#)

原文链接: <https://ld246.com/article/1559101819464>

来源网站: 链滴

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

具体问题为，在使用WebDriver正常登录网址后，进行了Cookie保存，但在下次自动反序列化加载Cookie时发现无法正常加载，导致每次需要重新登录验证，目前采用了一种曲线救国的方式，在序列化Cookie时调整域名为xxx.com，具体方案如下。



具体问题大家也可以参见：[Add cookie exception when dot exists before domain name](#)

首先，在成功登录后序列化Cookie时：

```
/**
 * 序列化CookieStore
 * @param driver WebDriver
 * @param cookieFile 序列化地址
 * @return 是否序列化成功
 */
public static boolean saveCookies(WebDriver driver, String cookieFile){
    try{
        Options options = driver.manage();
        Set<Cookie> cookieSet = options.getCookies();

        CookieStore cookieStore = CookieUtils.driverCookie2HttpCookie(cookieSet);

        return ObjectUtils.serializeFile(cookieStore, cookieFile);
    }catch(Exception e){
        e.printStackTrace();
        return false;
    }
}

/**
 * 将获取的DriverCookie值转换为HttpCookie
 * @param driver WebDriver
 * @return CookieStore
 */
```

```

*/
public static CookieStore driverCookie2HttpCookie(Set<org.openqa.selenium.Cookie> cookie
){
    CookieStore store = new BasicCookieStore();

    for (org.openqa.selenium.Cookie cookie : cookies) {
        BasicClientCookie bcc = new BasicClientCookie(cookie.getName(), cookie.getValue());

        // log.debug "[" + cookie.getName() + "]=[" + cookie.getValue() + "]" + cookie.getDomain() == null ? "" : cookie.getDomain());

        // 整理Cookie, 将www去掉, 否则Driver在addCookies时将会自动在前面加., 即www.abc.c
        m变成.www.abc.com, 从而无法使用
        String domain = cookie.getDomain();
        bcc.setDomain(StringUtils.isEmpty(domain) ? null : domain.startsWith("www") ? domain.
        substring(3) : domain);

        bcc.setPath(cookie.getPath().equals("/") ? "/" : cookie.getPath());
        bcc.setExpiryDate(cookie.getExpiry());

        store.addCookie(bcc);
    }

    return store;
}

```

然后就可以在下次加载WebDriver时正常加载Cookie了

```

/**
 * 给指定的URL填充Cookie
 * @param url 要加载的地址
 */
protected void addCookies(String url) {
    driver.get(url);

    if(null != cookieStore){
        driver.addCookies(cookieStore);
        driver.refresh();
        driver.waitPageLoad();
    }
}

/**
 * 给Driver增加Cookie
 * @param driver
 * @return
 */
public void addCookies(CookieStore cookieStore) {
    Options options = driver.manage();
    List<Cookie> cookieList = CookieUtils.httpCookie2DriverCookie(cookieStore);
    for(Cookie cookie : cookieList){
        // 整理Cookie, 将www去掉, 否则回填时将会自动在前面加., 即www.abc.com, 将变成.www
        abc.com, 而无法使用
        // 一般情况下在序列化时已经解决了前置www的问题, 在这儿应该就不需要再验证了, 但考虑

```

部分已经存在的CookieStore还存在前置www的问题，所以可以应用验证

```
// cookie = new Cookie(cookie.getName(), cookie.getValue(),  
//     CookieUtils.modifyDomain4Driver(cookie.getDomain()), cookie.getPath(), cookie.g  
tExpiry(), false, false);  
  
    options.addCookie(cookie);  
}  
}
```

到这儿，可能有人要问了，序列化时干嘛转来转去的，把Set<Cookie>转换为CookieStore，纯粹CookieStore有其它用处，你也可以不转换直接序列化。

另外，注意在WebDriver加载Cookie前需要先打开网址，不能直接先加载Cookie，Cookie要加载到个域名下，因此，需要先打开网址，再加载Cookie，加载完成后refresh刷新页面，最后通过规则指是否加载（登录）成功。

参见：<https://github.com/SeleniumHQ/selenium/issues/4861>