链滴

session 原理以及 session 共享

作者：614756773

原文链接：https://ld246.com/article/1589524113976

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)

cookie

说session之前一定离不开cookie

cookie是存储在浏览器中的一小段文本格式如：

a=xxx;b=xxxx;c=xxx

浏览器

浏览器端在发送请求时可能会带上cookie，如下：

```
Request Headers
:authority: www.cnblogs.com
:method: GET
:path: /ajianbeyourself/ajax/CommentForm.aspx?postId=4900140&_=1589522080185
:scheme: https
accept: text/html, */*; q=0.01
accept-encoding: gzip, deflate, br
accept-language: zh-CN,zh;q=0.9,zh-TW;q=0.8
cookie: _ga=GA1.2.958031816.1554176988; __gads=ID=c05e695fe0ecf4df:T=1554176998:S=
LNI_MYjFCirz0z4LlEeAjPh8DENgWhU2w;
referer: https://www.cnblogs.com
```

服务端

有些cookie是浏览器端主动生成，然后在请求服务端时发送的，还有一些是在服务端生成然后告知给览器的

当服务端需要告诉浏览器存储并且往后的所有请求都带上某个cookie时，会在response中添加如下he

der:

Set-Cookie:value [;expires=date][;domain=domain][;path=path][;secure]

[ ]中的值时可选项，value是必选的，入之前所述value的格式一般是a=xxx;b=xxxx但却不是必须的

当浏览器发现响应头中有Set-Cookie这个header那么就会把cookie存储，以后对该服务器的所有请都会带上这个cookie

---

session

session是和cookie配合使用的，可以这么说session就是用一个map的数据结构，而cookie就是key

一般用户登录后，服务器会存储一些登录信息在缓存中，然后生成一个cookie其实就是一个key返回前端，前端以后访问时会带上这个key其实就是带上cookie，这个是浏览器自动帮我们做了的操作，务端就可以对这次请求进行一些检查操作了，比如说检查用户是否已登录

session共享

传统的session是直接存储在内存中的，如果是单机部署应用肯定没问题，集群就不行了

比如说小明在登录时，被负载均衡后交由server A处理，server A处理后会把session存储在本机的内中，然后返回登录成功

接着，小明在网站进行一些其他操作，比如说查看好友列表，此时这个请求被负载均衡器交由server处理，server B拿到cookie，然后根据cookie去查找session，肯定找不到于是就认为小明没有登录返回400。。

针对这种问题有很好的轮子可以解决：Spring Session

这个问题出现的根本原因是各个服务器都把session缓存在自己的内存中，所以只需要把session集中一个地方存储就行了，Spring Session就是这样做的。

Spring Session常用的一种方案是把session存储在redis中，各个服务器公用同一个redis

spring session的原理以及设计可以看我另一篇文章

内部服务调用session共享

上面的问题只是解决了客户端调用服务时session可以共享

server A1，server A2，server A3是一个服务集群，提供相同的服务
小明登录后，请求任意一个server都没问题

现在有了新的问题，则是内部服务之间的调用，比如

有服务server A和server B，他们提供不同的服务
小明通过server A进行登录，然后查看好友列表
`查看好友列表`这个操作需要客户端调用server A，然后server A调用server B

在server A中肯定能够拿到小明登录后的信息，但是server B则不行。

解决办法很简单，如下：

- server A和server B使用 spring session，公用同一个redis存储session
- server A调用server B时，把cookie取出来然后传递给server B

之前说过了cookie就是key，server B有了cookie肯定能拿到session的值了

这是spring session帮我们做的，虽然我没仔细看过源码，但是其步骤肯定如下：

- 检查请求头中是否有cookie
- 检查是否有 SESSION=ABCDEFGHXXXX的cookie
- 若有则根据 ABCDEFGHXXXX去redis中查询数据，然后存放在HttpSession中