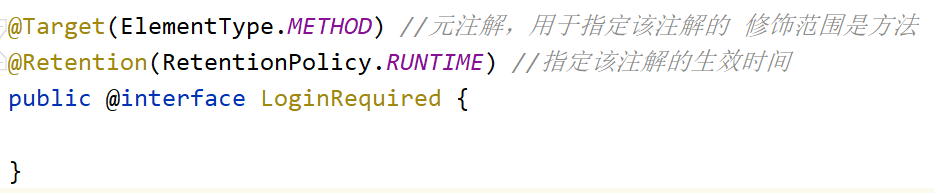
## 注解是怎么实现的？

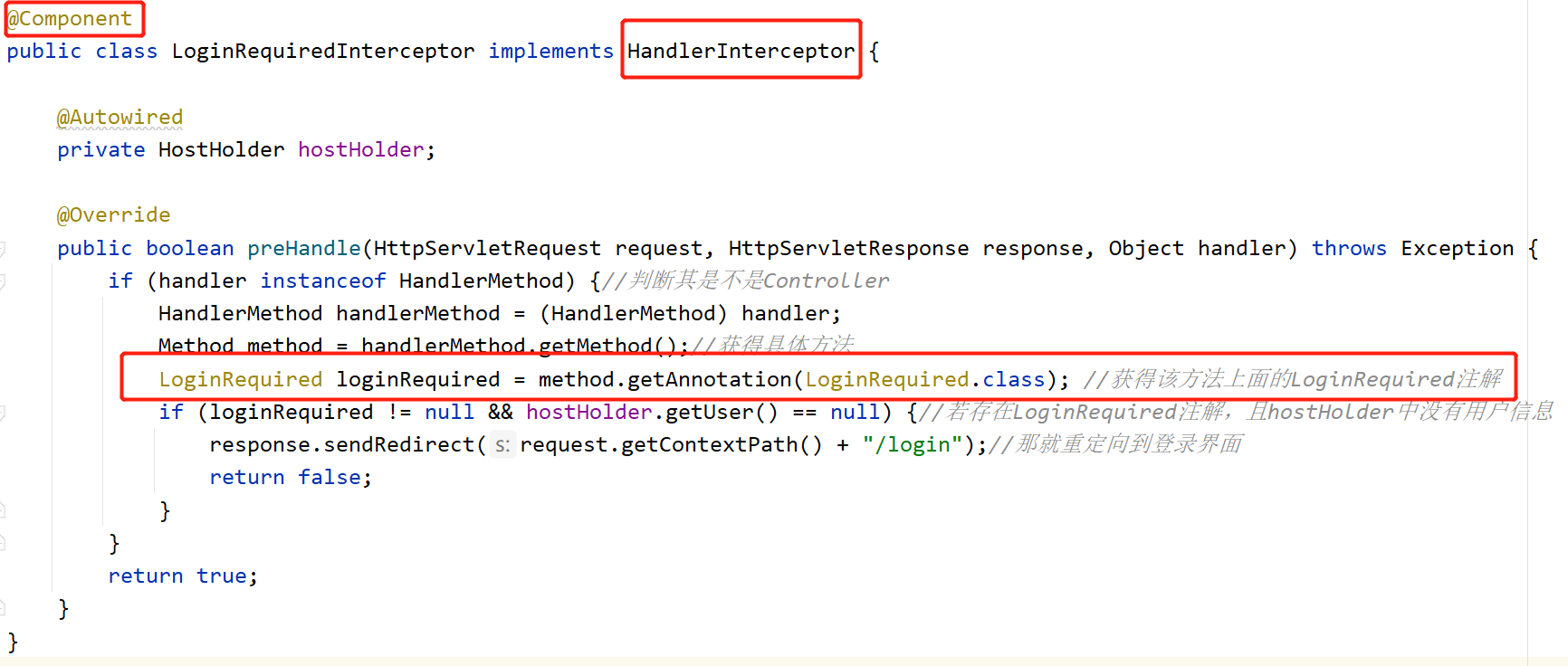
注解其实是代码里面的特殊标记，这些标记可以在编译、类加载、运行时被读取，并执行相应的处理。

注解是一个接口，程序可以通过反射来获取制定程序元素的Annotation对象，以此来取得注解里面的元数据。

自定义注解：



拦截器，拦击带有LoginRequired注解的controller方法。带有此注解意味着需要登录检查。



## ThreadLoca+Redis替代Session，在线程内保存用户信息



对一个请求的处理是在一个线程内实现的。

因此可以使用HostHolder保存用户信息，使用拦截器，在访问到达controller之前就根据请求中的cookie中的sessionid查找redis中对应的用户信息（包含用户名、过期时间等等），保存在Hostholder中，因此不论是在controller还是service处需要使用用户信息的时候，就直接注入HostHolder对象即可。

且该对象内的成员变量是ThreadLocal，因此，对应不同的线程都是不同的用户。

这样就避免了每次需要用户信息就去Redis中查找了。也不需要想Service中单独传入用户信息或者sessionid了。