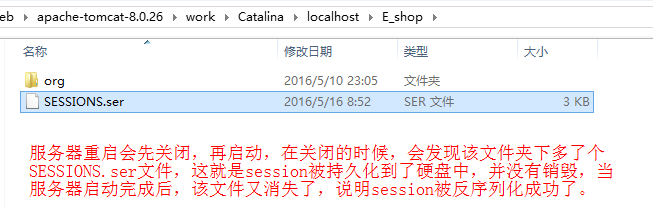
最近在写网上商城项目的时候学习了一个关于session的序列化问题，过来总结一下。

        众所周知，session是服务器端的一种会话技术，只要session没有关闭，一个会话就会保持。这里先引出一个问题：**如果我在访问某个页面后，服务器重启了一下，但是网页还没关，那么原来的session还在么**？答案是很明显的，你都把服务器关掉了，session肯定不是原来的session了，原来的像登录信息等一些跟session相关的信息肯定就没了。但是如果我们想要服务器重启后，还是原来的session，那跟如何做呢？

        这就涉及到了一个叫序列化（Serializable）的技术。当对象存储到硬盘的时候，就需要实现序列化接口，序列化的功能就是添加了一个唯一的ID（类主键），这样在反序列化（从硬盘加载到内存）的时候就可以成功找到相应的对象。另外，还要弄清楚一件事情：一般大家都觉得容器关闭后，session就销毁了，其实不是这样的，**容器的关闭并不会导致session的销毁**。过程是这样子的，一旦容器关闭后，session就会被持久化到硬盘，并没有真正销毁，为了说明这个问题，来做个试验：打开tomcat的工作目录下正在运行的工程目录：我的是E:\web\apache-tomcat-8.0.26\work\Catalina\localhost\E\_shop，里面只有一个org的文件夹，其他什么也没有，现在我们重启tomcat服务器，注意观察这里面的变化，当服务器停掉后，这个该目录下多了个SESSION.ser文件，服务器重启成功后，该文件又消失了。如下：



        所以，如果项目中的POJO实现了Serializable接口，当反序列化的时候就能找到刚刚序列化时候的POJO，原来session中的内容就能成功反序列化，session还是原来的session，这样原来页面的东西还在，刷新后还是继续上次的操作。如果POJO没有被实例化，那么在session发序列化的时候当然就没有了这些POJO了。下面看一下我的项目中的部分POJO，如下：



        最后总结一下：

        1. 容器关闭后session并没有消失，而是被持久化到了硬盘里；

        2. 如果项目中的POJO实现了Serializable接口，那么会跟着session一起被持久化到硬盘，在反序列化的时候会成功还原；

        3. 要想服务器重启后，还是原来的session，还继续紧接着原来的页面操作的话，就需要实例化项目中的POJO。

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

-----乐于分享，共同进步！

-----我的博客主页：<http://blog.csdn.net/eson_15>