原创 session绑定与解绑，钝化活化

2019-10-15 17:29:09 Zero-place 阅读数 48 更多

本文链接：https://blog.csdn.net/qq_41877184/article/details/102571265

session对象的四种状态：
1）绑定、解绑
2）与钝化(序列化、持久化)、活化(反序列化)
二者的联系：对象绑定在session中(监听1)，session钝化对象随之钝化(监听2)。

下面将介绍这两个过程的监听即：绑定、解绑 与钝化、活化。

session绑定和解绑(无需配置web.xml)：

监听象需实现：HttpSessionBingListener接口。
作用： javabean实现HttpSessionBingListener接口后，javabean就知道自己是否添加到了session中，是否被session移除。
1.将对象a[绑定]在session中：session.setAttribute("a",xxx)
2.将对象a从session中[解绑]：session.removceAttribute("a")

测试代码：

```
public class BeanListener implements HttpSessionBindingListener {
        @Override
        //绑定
        public void valueBound(HttpSessionBindingEvent arg0) {
                System.out.println("绑定bean对象(将bean对象增加到session域中)，bean对象地
        this+",sessionId:"+arg0.getSession().getId());
        }

        @Override
        //解绑
        public void valueUnbound(HttpSessionBindingEvent arg0) {
                System.out.println("解除bean对象(将bean从session域中移除)，bean对象地址：
        this+",sessionId:"+arg0.getSession().getId());
        }
}
```

```
<body>
        <%
                BeanListener bean=new BeanListener();
            session.setAttribute("bean", bean);//绑定
            session.removeAttribute("bean");//解绑
        %>

</body>
```

运行代码2次结果如下(即访问该.jsp，再刷新一次)：

```
绑定对象(对象增加到session域)，bean对象地址：org.threetierarc.listener.BeanListener@165f
解除对象(从session域中移除)，bean对象地址：org.threetierarc.listener.BeanListener@165f6d
绑定对象(对象增加到session域)，bean对象地址：org.threetierarc.listener.BeanListener@4af3
解除对象(从session域中移除)，bean对象地址：org.threetierarc.listener.BeanListener@4af347
```

结果分析： 第一次创建bean1对象，进行session绑定，然后解除，第二次创建新的bean2对象，进行session绑定，然后再解除，因此bean地址不同，而sessionid相同。

session钝化与活化(无需配置web.xml)：

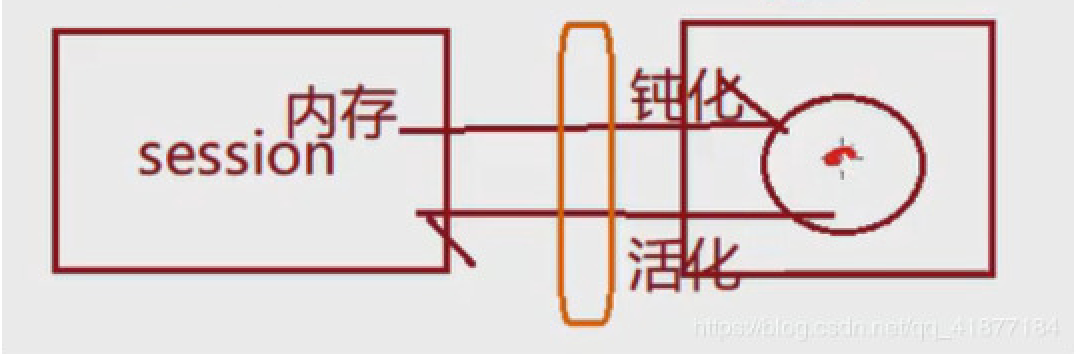
理解钝化与活化：
钝化(序列化)：内存--->硬盘
活化(反序列化)：硬盘--->内存

作用：
情况一： 服务器内存有限，在很多用户访问服务器时，存在很多session，为了节省资源，将长时间没有活动的用户session信息序列化存入硬盘，当用户再次活动时，再从硬盘中反序列化调回session信息，用户没有察觉。
情况二： 在万不得已的情况下重启，那么为了用户体验，必须将session信息保存在硬盘中，当重启后，session在回到内存。
因此总结：保存session到硬盘，需要保留用户的session信息，该用户的信息保存在javabean中，此监听接口就是用来监听该javabean随session一起保存到硬盘了没。

监听session的钝化、活化实现： HttpSessionActivationListener接口，Serializable接口
监听代码：

```
package org.threetierarc.listener;

import javax.servlet.http.HttpSessionActivationListener;
import javax.servlet.http.HttpSessionEvent;

public class Bean implements HttpSessionActivationListener,Serializable{
        private String username;
        private String password;

        public Bean(){

        }
        public Bean(String username, String password) {
                super();
                this.username = username;
                this.password = password;
        }
        public String getUsername() {
                return username;
        }
        public void setUsername(String username) {
                this.username = username;
        }
        public String getPassword() {
                return password;
        }
        public void setPassword(String password) {
                this.password = password;
        }
        //监听时刻：活化之后
        @Override
        public void sessionDidActivate(HttpSessionEvent arg0) {
                // TODO Auto-generated method stub
                System.out.println("Bean即将与session一起：活化");
        }
        //监听时刻：钝化之前
        @Override
        public void sessionWillPassivate(HttpSessionEvent arg0) {
                System.out.println("Bean即将与session一起：钝化");

        }


}
```

特别注意：监听时刻->钝化前与活化后



完成钝化、活化操作：

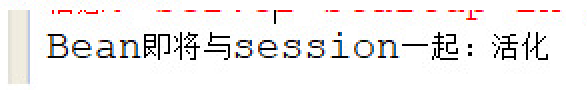
配置的方式主要有3种，3选一即可。
（1）在tomcat里面 conf/context.xml 里面配置 (我选的)
特点：对所有的运行在这个服务器的项目生效
（2）在conf/Catalina/localhost/context.xml 配置
特点：对 localhost生效。 localhost:8080
（3）在自己的web工程项目中的 META-INF/context.xml
特点：只对当前的工程生效。
配置文件：

```
IdleSwap:最大空闲时间，如果超过该时间，将会被钝化
eStore: 通过该类 具体实现 钝化操作
ectory:相对路径(相对于Tomcat安装目录/work/Catalina/localhost/项目名，
者eclipse的G:\eclipse\.metadata\.plugins\org.eclipse.wst.server.core\tmp1\work\Catalina\loc
可以写绝对路径。



- session对象5秒没有访问就钝化 -->
 <Manager className="org.apache.catalina.session.PersistentManager" maxIdleSwap="5">
        <Store className="org.apache.catalina.session.FileStore" directory="lq"></Store>
 </Manager>
```

测试jsp代码：

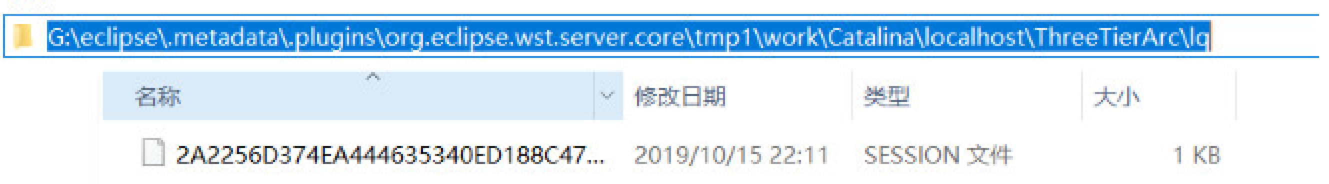
```
        <%
                Bean bean=new Bean("wang","123456");
            session.setAttribute("bean", bean);
        %>
```

控制台输出：



```
Bean即将与session一起：活化
```

钝化文件：



G:\eclipse\.metadata\.plugins\org.eclipse.wst.server.core\tmp1\work\Catalina\localhost\ThreeTierArc\lq

名称	修改日期	类型	大小
2A2256D374EA444635340ED188C47...	2019/10/15 22:11	SESSION 文件	1 KB

过程分析： 启动项目，运行jsp代码后，不对session进行操作，5秒过后将进行钝化，控制台输出"即将钝化"，成功钝化后生成.session文件保存在硬盘中，若关闭服务器，再次启动服务器，将要对原有session进行操作，则现在内存中寻找该session，如果没有则在硬盘中找，如果找到则自动反序列化，即钝化文件会销毁并回到内存！