session

session机制采用的是在服务器端保持HTTP状态信息的方案。 session底层默认是用一个cookie来操作，表示客户端与服务端的一次会话，客户端向服务器端发送请求时，服务器端会检测本次会话的cookie中，是否包含name为JSESSIONID的cookie，如果没有包含则创建一个cookie发送到客户端。会话结束则cookie消亡，作为本次会话的唯一标示。可以通过手动设置用URL重写的方式保存session，最下面有介绍

我们可以手动的把session cookie持久化

持久化session cookie

Cookie cookie=**new** Cookie("JSESSIONID",session.getId());

cookie.setMaxAge(5);

response.addCookie(cookie);

session.setMaxInactiveInterval(20); //设置过期时间

1. HttpSession的生命周期
2. 什么时候创建session
3. 是否浏览器访问服务端的任何一个jsp或者servlet，服务器都会创建一个HttoSession对象呢？不一定。
   * 1. 若当前jsp是浏览器访问的第一个资源，如果该jsp的page中设置session=false则服务器不会为jsp创建session对象
     2. 若当前jsp不是浏览器访问的第一个资源，在此之前已有jsp创建了session对象，则会把和当前会话关联的session对象返回给jsp
     3. 对于servlet而言，如果当前servlet是浏览器访问的第一个资源，只有在调用request.getSession()或

request.getSession(true)时会创建

再次之外都会创建session

1. session=”false”是什么意思？

当前jsp内不能使用sessio隐含对象，但可以使用其他显示的HttpSession对象 例，request.getSession(false); 如果设置为false可以用request.getSession()方法获取使用session。

1. request.getSession()方法

**如果不带参数或者参数为true，**

* + 1. 如果当前jsp或servlet没有session，会创建一个session并返回
    2. 如果有与当前jsp或servlet关联的session，则返回已关联的这个对象

**如果参数为false**

1. 如果当前jsp或servlet没有session，返回null
2. 如果有与当前jsp或servlet关联的session，则返回已关联的这个对象
3. 销毁session对象
   1. 直接调HttpSession的invalidate()方法销毁

会马上使当前session对象失效，销毁后再次发送请求会按照session创建的规则重新创建

* 1. 服务器卸载当前web应用（web服务器进程被终止）
  2. 超过session的过期时间

在不向web服务器发出请求时session的存活时间

session.setMaxInactiveInterval(秒); //设置过期时间

此方法用来设置session的过期时间，默认为1800秒（半个小时）

session.getMaxInactiveInterval(); //获取过期时间

在xml文件中设置session默认的过期时间

<session-config>

<session-timeout>30</session-timeout>

</session-config>

* 1. 并不是关闭了浏览器就session就被销毁，还在服务器里存着，等待被召唤，只是sessionc cookie被销毁，通过方法可以再次唤醒该session。

1. 相关的API
   1. 获取session对象：request.getSession、

request.getSession(Boolean create)

* 1. 属性相关：setAttribute，getAttribute，removeAttribute
  2. 使HttpSession失效的：invalidate()
  3. 设置其最大失效的：setMaxInacriveInterval

**补充：**

1. url重写

在跳转时，对url进行操作，

例hello.jsp 改为response.encodeURL(“hello.jsp”)

或response.encodeRedirectURL(“hello.jsp”) //二者无区别

在跳转时会把本次session的id缀在url后面

例hello.jsp:jsessionid=(当前sessionid)

这样的操作，使session从用cookie来获取，改为用url的方式来获取避免cookie被禁用无法使用session

1. 避免表单重复提交
   1. 重复提交的情况
   2. 在表单提交到一个servlet转发到另一个页面，此时地址栏路径依然是servlet的路径，点击刷新，出现重复提交
   3. 在表单未完成提交时，重复点击提交按钮，出现重复提交
   4. 表单完成提交后，但会提交前页面重新提交，出现重复提交
   5. 不是重复提交

提交后“返回”，“刷新”原页面，在提交

* 1. 解决方法

设置一个标志值（最好是随机的复杂值）放session属性中，

例new Date().getTime()+””

* + - 1. 把这个值放入属性里name为token
      2. 把这个值放到表单的一个隐藏域里name为token
      3. 提交后，比较session属性token和提交的请求参数token是否一致，如果一致，则移除该session属性，执行。如果不一致就证明不是从原表单过来的，提示重复提交，并结束程序。

1. 验证码