# 凯讯科技

## session的工作原理，有什么好处？

在java WEB开发中，session既是重点又是难点，很多初学者都没有建立一个清晰的概念，希望通过下面这个示例对大家有所帮助。

在一个JavaWeb应用程序中新建2个jsp页面，名称分别为first.jsp和second.jsp。都是如下代码：  
< %@ page language="java" pageEncoding="gb2312"%>  
< html>  
 < head>    
   < title>First.jsp page</title>  
  </head>  
  <body>  
  <%  
    out.println(session.getId()+”<br>”);  
    out.println(session.isNew());  
  %>  
  </body>  
< /html>  
然后我们启动服务器在同一个浏览器页面中先访问first.jsp然后再访问second.jsp，结果如下：  
first.jsp  
60FF62D46CCB9088DA4BA6125E612ED4  
true  
second.jsp  
60FF62D46CCB9088DA4BA6125E612ED4  
false  
可以看到Id号码是一样的，当我们第一次访问一个Jsp页面时session对象就会创建，而以后的其他页面都是引用相同的session对象。  
打开两个浏览器页面，分别访问first.jsp和second.jsp我们可以看到这时两个页面中的id号是不一样的，而且都是true，我们可以很容易得出结论， 在相同的浏览器页面中是共享同一个session对象的。在大多数书籍中都是这样介绍session作用域的。但是情况并非全部如此，多个浏览器页面其实是完全可以共享相同的session对象的。下面我们慢慢解开谜底。  
当我们打开浏览器页面第一次访问first.jsp 页面时，容器会生成一个session对象（设置了<%@ page session=false %>指令除外），并将这个session对象的Id号码通过cookie的形式传送到客户端（但是这个cookie不会写入到客户端硬盘上，通常是存在浏览器的缓存当中，当这个页面关闭后它就随之消失，在IE中这个cookie的设置叫做会话cookie，可以尝试一下将这个会话cookie禁用掉，结果会如何），这个cookie的名称一般称为jsessionid，值当然是sesion的id号码。当客户通过此浏览器页面再次访问该web应该程序的其他页面时，就会将这个cookie传送给容器，容器就会通过这个jsessionid号码找到已经生成session对象.如果当发送的请求没有与之相关联的jsessionid时，容器才会生成一个新的session对象。如果客户端禁用掉了会话cookie那么容器就不会根据请求找到任何已经建立的session对象，所以每次都会生成新的session对象，这样会对容器的性能大打折扣，所以在这种情况下可以设置session指令，告诉容器不必为每一个页面都生成一个新的session对象。但是这种方式有一定的局限性，除非每个页面都不使用session对象，但是session的功能早已经深入人心，这种情况我们可以使用重写URL的方式，手工的给其他的页面传递jsessionid号码，这样容器就能够通过传递的id号码找到以前的 session对象，而不会再生成新的session对象了。这种方式完全可以打破同一个浏览器页面共享一个session对象的说法，多个浏览器页面也完全可以。

## 虚拟机的工作流程和原理？

## 三、全局变request、pagecontext、。。。。九个？

## 四、一座大桥 a:1分钟 b：2分钟 c、5分钟 d：10分钟

## 17分钟 之内 必须过桥 如果 分配，一次只能两个人过桥。

## 五、system.gc().

## 六、static有哪些作用。如果 并发执行时用了static 会如何？不会并发执行。

## 七、hasmap安全吗？

## 八、实现线程有几种方式？

## 继承Thread 接口Runable 和 callable

# 面试总结

## 一、String是最基本的数据类型吗?

String不是基本的数据类型。

基本的数据类型只有8种，byte、short、int、long、char、float、double、boolean，

基本数据类型仅仅只有8种。到1.8以后有增加了一种void。

## 二、&和&&的区别？

&是位运算符，表示按位与运算，&&是逻辑运算符，表示逻辑与（and）。

## 三、HashMap和Hashtable的区别？

HashMap是Hashtable的轻量级实现（非线程安全的实现），他们都完成了Map接口，主要区别在于HashMap允许空（null）键值（key）,由于非线程安全，效率上可能高于Hashtable。

HashMap允许将null作为一个entry的key或者value，而Hashtable不允许。

Hashtable继承自Dictionary类，而HashMap是Java1.2引进的Map?interface的一个实现。

不同是，Hashtable的方法是Synchronize的，而HashMap不是，在多个线程访问Hashtable时，不需要自己为它的方法实现同步，而HashMap?必须为之提供外同步。

## 四、sleep()?和?wait()?有什么区别??

sleep是线程类（Thread）的方法，导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复。调用sleep不会释放对象锁。

wait是Object类的方法，对此对象调用wait方法导致本线程放弃对象锁，进入等待此对象的等待锁定池，只有针对此对象发出notify方法（或notifyAll）后本线程才进入对象锁定池准备获得对象锁进入运行状态。

## 五、heap和stack有什么区别？

栈是一种线形集合，其添加和删除元素的操作应在同一段完成。栈按照后进先出的方式进行处理。堆是栈的一个组成元素

## 六、如何创建线程？

◆需要从Java.lang.Thread类派生一个新的线程类，重载它的run()方法；

◆实现Runnalbe接口，重载Runnalbe接口中的run()方法。