

CSDN

联系我们

请扫描二维码联系客服

webmaster@csdn.net

400-660-0108

QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

等级： 博客 3

访问量：4万+

积分：696

排名：7万+

文章搜索

文章分类

JAVA (93)

JAVA并发编程 (16)

数据库 (25)

sql (12)

mysql (16)

spring (14)

JAVA, ZK (1)

正则 (1)

servlet (3)

web (1)

移动 (1)

前端开发 (1)

mybatis (2)

JAVA, 网络编程 (1)

缓存 (2)

Ehcache (1)

网络编程 (3)

NIO (1)

同步异步 (1)

Oracle (5)

redis (9)

并发 (2)

GitChat 论坛 ...

写博客

发Chat

目录视图

摘要视图

0 订阅

Servlet是单例的 所以需要线程安全 以及如何实现线程安全

标签：java servlet 多线程

2017年03月15日 09:41:41

1255人阅读

评论(0)

收藏

举报

分类：

JAVA (92) JAVA并发编程 (15) servlet (2)

Servlet/JSP技术和ASP、PHP等相比，由于其多线程运行而具有很高的执行效率。由于Servlet/JSP默认是以多线程模式执行的，所以，在编写代码时需要非常细致地考虑多线程的安全性问题。然而，很多人编写Servlet/JSP程序时并没有注意到多线程安全性的问题，这往往造成编写的程序在少量用户访问时没有任何问题，而在并发用户上升到一定值时，就会经常出现一些莫名其妙的问

题。

Servlet的多线程机制

Servlet体系结构是建立在Java多线程机制之上的，它的生命周期是由Web容器负责的。当客户端第一次请求某个Servlet时，Servlet容器将会根据web.xml配置文件实例化这个Servlet类。当有新的客户端请求该Servlet时，一般不会再实例化该Servlet类，也就是有多个线程在使用这个实例。Servlet容器会自动使用线程池等技术来支持系统的运行，如图1所示。

Servlet容器

线程

线程

Servlet线程池

客户端

分配一个线程处理请求

分配一个线程处理请求

这样，当两个或多个线程同时访问同一个Servlet时，可能会发生多个线程同时访问同一资源的情况，数据可能会变得不一致。所以在用Servlet构建的Web应用时如果不注意线程安全的问题，会使所写的Servlet程序有难以发现的错误。

Servlet的线程安全问题

Servlet的线程安全问题主要是由于实例变量使用不当而引起的，这里以一个现实的例子来说明。

Import javax.servlet. *;
Import javax.servlet.http. *;
Import java.io. *;
Public class Concurrent Test extends HttpServlet {PrintWriter output;
Public void service (HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {String username;
Response.setContentType ("text/html; charset=gb2312");
Username = request.getParameter ("username");
Output = response.getWriter ();
Try {Thread. sleep (5000); //为了突出并发问题，在这设置一个延时
} Catch (InterruptedException e){
output.println("用户名:"+Username+"
");

http://blog.csdn.net/i_will_try/article/details/62215633

1/11

秒杀 (5)

联系我们



请扫描二维码联系客服
webmaster@csdn.net
400-660-0108
QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

数据库连接池 (1)

mq (1)

springmvc (2)

框架 (1)

json (1)

限流 (1)

悲观锁 (2)

乐观锁 (4)

日志 (1)

QPS (1)

分表 (1)

临时表 (3)

锁 (2)

redisTemplate (1)

定时任务 (1)

ThreadLocal (1)

web开发 (1)

分布式锁 (1)

watch (1)

事务 (2)

springboot (10)

session (2)

seo (1)

源码 (1)

编辑器 (2)

Intellij idea (3)

storm (1)

面试 (1)

scala (1)

hash (2)

算法 (1)

sso (2)

cookie (1)

HashMap (3)

jvm (1)

生活 (2)

二叉树 (1)

数据结构 (1)

quartz (1)

文章存档

2017年12月 (15)

```
}  
}
```

该Servlet中定义了一个实例变量output，在service方法将其赋值为用户的输出。当一个用户访问该Servlet时，程序会正常的运行，但当多个用户并发访问时，就可能会出现其它用户的信息显示在另外一些用户的浏览器上的问题。这是一个严重的问题。为了突出并发问题，便于测试、观察，我们在用户信息时执行了一个延时的操作。假设已在web.xml配置文件中注册了该Servlet，现有两个用户a和b同时访问该Servlet（可以启动两个IE浏览器，或者在两台机器上同时访问），即同时在浏览器中输入：

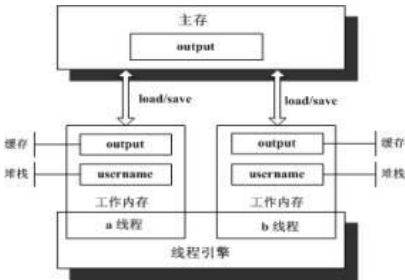
- a: http://localhost: 8080/servlet/ConcurrentTest? Username=a
- b: http://localhost: 8080/servlet/ConcurrentTest? Username=b

如果用户b比用户a回车的时间稍慢一点，将得到如图2所示的输出：



从图2中可以看到，Web服务器启动了两个线程分别处理来自用户a和用户b的请求，但是在用户a的浏览器上却得到一个空白的屏幕，用户a的信息显示在用户b的浏览器上。该Servlet存在线程不安全问题。下面我们就从分析该实例的内存模型入手,观察不同时刻实例变量output的值来分析使该Servlet线程不安全的原因。

Java的内存模型JMM（Java Memory Model）JMM主要是为了规定了线程和内存之间的一些关系。根据JMM的设计，系统存在一个主内存(Main Memory)，Java中所有实例变量都储存在主存中，对于所有线程都是共享的。每条线程都有自己的工作内存(Working Memory)，工作内存由缓存和堆栈两部分组成，缓存中保存的是主存中变量的拷贝，缓存可能并不总与主存同步，也就是缓存中变量的修改可能没有立刻写到主存中；堆栈中保存的是线程的局部变量，线程之间无法相互直接访问堆栈中的变量。根据JMM，我们可以将论文中所讨论的Servlet实例的内存模型抽象为图3所示的模型。



下面根据图3所示的内存模型，来分析当用户a和b的线程（简称为a线程、b线程）并发执行时，Servlet实例中所涉及变量的变化情况 & 线程的执行情况，如图4所示。

调度时刻	a线程	b线程
T1	访问Servlet页面	
T2		访问Servlet页面
T3	output=a的输出username=a休眠5000毫秒，让出CPU	

2017年11月 (28)

联系我们



请扫描二维码联系客服
webmaster@csdn.net
400-660-0108
QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

SQL习题笔记 (809)

ZK框架学习 (699)

最新评论

聊聊Dubbox (二) : 简单入门
L_小乌鸦 : [reply]weixin_37519367[/reply] 加了这个依赖吗? 这种情况我也没遇见过...

聊聊Dubbox (二) : 简单入门
呵呵小短腿 : 博主, 我的报这个错: Could not find MessageBodyWriter for r...

mysql存储过程详解
ET_0001 : 可以, 非常详细~谢谢博主

T4		output=b的输出（写回主存）username=b休眠5000毫秒，让出CPU
T5	在用户b的浏览器上输出a线程的username的值,a线程终止。	
T6		在用户b的浏览器上输出b线程的username的值,b线程终止。

图4 Servlet实例的线程调度情况

从图4中可以清楚的看到，由于b线程对实例变量output的修改覆盖了a线程对实例变量output的修改，从而导致了用户a的信息显示在了用户b的浏览器上。如果在a线程执行输出语句时，b线程对output的修改还没有刷新到主存，那么将不会出现图2所示的输出结果，因此这只是一种偶然现象，但这更增加了程序潜在的危险性。

设计线程安全的Servlet

通过上面的分析，我们知道了实例变量不正确的使用是造成Servlet线程不安全的主要原因。下面针对该问题给出了三种解决方案并对方案的选取给出了一些参考性的建议。

1、实现 SingleThreadModel 接口

该接口指定了系统如何处理对同一个Servlet的调用。如果一个Servlet被这个接口指定,那么在这个Servlet中的service方法将不会有两个线程被同时执行，当然也就不存在线程安全的问题。这种方法只要将前面的Concurrent Test类的类头定义更改为：

```
Public class Concurrent Test extends HttpServlet implements SingleThreadModel {
.....
}
```

2、同步对共享数据的操作

使用synchronized 关键字能保证一次只有一个线程可以访问被保护的区段，在本论文中的Servlet可以通过同步块操作来保证线程的安全。同步后的代码如下：

```
.....
Public class Concurrent Test extends HttpServlet { .....
Username = request.getParameter ("username");
Synchronized (this){
Output = response.getWriter ();
Try {
Thread. Sleep (5000);
} Catch (InterruptedException e){}
output.println("用户名:"+Username+"<BR>");
}
}
}
```

3、避免使用实例变量

本实例中的线程安全问题是实例变量造成的，只要在Servlet里面的任何方法里面都不使用实例变量，那么该Servlet就是线程安全的。

修正上面的Servlet代码，将实例变量改为局部变量实现同样的功能，代码如下：

```
.....
Public class Concurrent Test extends HttpServlet {public void service (HttpServletRequest request, HttpServletResponse) throws ServletException, IOException {
Print Writer output;
```

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
💬 QQ客服 💬 客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

```
String username;  
Response.setContentType("text/html; charset=gb2312");  
.....  
}  
}
```

对上面的三种方法进行测试，可以表明用它们都能设计出线程安全的Servlet程序。但如果一个Servlet实现了SingleThreadModel接口，Servlet引擎将为每个新的请求创建一个单独的Servlet实例，这将引起大量的系统开销。SingleThreadModel在Servlet2.4中已不再提倡使用；同样如果在程序中使用要使用的共享的数据，也会使系统的性能大大下降。这是因为被同步的代码块在同一时刻只能有一个线程执行它，使得其同时处理客户请求的吞吐量降低，而且很多客户处于阻塞状态。另外为保证主存中线程的工作内存中的数据的一致性，要频繁地刷新缓存,这也会大大地影响系统的性能。所以在实际的开发中也应避免或最小化 **Servlet 中的同步代码；在Servlet中避免使用实例变量是保证Servlet线程安全的最佳选择。**从Java 内存模型也可以知道，方法中的临时变量是在栈上分配空间，而且每个线程都有自己私有的栈空间，所以它们不会影响线程的安全。

小结

Servlet的线程安全问题只有在大量的并发访问时才会显现出来，并且很难发现，因此在编写Servlet程序时要特别注意。线程安全问题主要是由实例变量造成的,因此在Servlet中应避免使用实例变量。如果应用程序设计无法避免使用实例变量，那么使用同步来保护要使用的实例变量，但为保证系统的最佳性能，应该同步可用性最小的代码路径。

一，Servlet容器如何同时处理多个请求。

Servlet采用多线程来处理多个请求同时访问，Servlet容器维护了一个线程池来服务请求。线程池实际上是等待执行代码的一组线程叫做工作者线程(Worker Thread)，Servlet容器使用一个调度线程来管理工作者线程(Dispatcher Thread)。

当容器收到一个访问Servlet的请求，调度者线程从线程池中选出一个工作者线程，将请求传递给该线程，然后由该线程来执行Servlet的service方法。
当这个线程正在执行的时候，容器收到另外一个请求，调度者线程将从池中选出另外一个工作者线程来服务新的请求，容器并不关系这个请求是否访问的是同一个Servlet还是另外一个Servlet。
当容器同时收到对同一Servlet的多个请求，那这个Servlet的service方法将在多线程中并发的执行。

二，Servlet容器默认采用单实例多线程的方式来处理请求，这样减少产生Servlet实例的开销，提升了对请求的响应时间。对于Tomcat可以在server.xml中通过<Connector>元素设置线程池中线程的数目。

就实现来说：
调度者线程类所担负的责任如其名字，该类的责任是调度线程，只需要利用自己的属性完成自己的责任。所以该类是承担了责任的，并且该类的责任又集中到唯一的单体对象中。
而其他对象又依赖于该特定对象所承担的责任，我们就需要得到该特定对象。那该类就是一个单例模式的实现了。

三，如何开发线程安全的Servlet

1，变量的线程安全：这里的变量指字段和共享数据(如表单参数值)。

a，将 参数变量 本地化。多线程并不共享局部变量.所以我们要尽可能的在servlet中使用局部变量。
例如：String user = "";
user = request.getParameter("user");

b，使用同步块Synchronized，防止可能异步调用的代码块。这意味着线程需要排队处理。
在使用同步块的时候要尽可能的缩小同步代码的范围，不要直接在service方法和响应方法上使用同步，这样会严重影响性能。

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
💬 QQ客服 💬 客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

2,属性的线程安全：ServletContext，HttpSession，ServletRequest对象中属性

ServletContext: （线程是不安全的）

ServletContext是可以多线程同时读/写属性的，线程是不安全的。要对属性的读写进行同步处理或者进行深度Clone()。

所以在Servlet上下文中尽可能少量保存会被修改（写）的数据，可以采取其他方式在多个Servlet中共享，比方我们可以使用单例模式来处理共享数据。

HttpSession: （线程是不安全的）

HttpSession对象在用户会话期间存在，只能在处理属于同一个Session的请求的线程中，因此Session对象的属性访问理论上是线程安全的。

当用户打开多个同属于一个进程的浏览器窗口，在这些窗口的访问属于同一个Session，会出现多次请求，需要多个工作线程来处理请求，可能造成同时多线程读写属性。

这时我们需要对属性的读写进行同步处理：使用同步块Synchronized和使用读/写器来解决。

ServletRequest: （线程是安全的）

对于每一个请求，由一个工作线程来执行，都会创建有一个新的ServletRequest对象，所以ServletRequest对象只能在一个线程中被访问。ServletRequest是线程安全的。

注意：ServletRequest对象在service方法的范围内是有效的，不要试图在service方法结束后仍然保存请求对象的引用。

3，使用同步的集合类：

使用Vector代替ArrayList，使用Hashtable代替HashMap。

4，不要在Servlet中创建自己的线程来完成某个功能。

Servlet本身就是多线程的，在Servlet中再创建线程，将导致执行情况复杂化，出现多线程安全问题。

5，在多个servlet中对外部对象(比方文件)进行修改操作一定要加锁，做到互斥的访问。

四，SingleThreadModel接口

javax.servlet.SingleThreadModel接口是一个标识接口，如果一个Servlet实现了这个接口，那Servlet容器将保证在一个时刻仅有一个线程可以在给定的servlet实例的service方法中执行。将其他所有请求进行排队。

服务器可以使用多个实例来处理请求，代替单个实例的请求排队带来的效益问题。服务器创建一个Servlet类的多个Servlet实例组成的实例池，对于每个请求分配Servlet实例进行响应处理，之后放回到实例池中等待下此请求。这样就造成并发访问的问题。

此时,局部变量(字段)也是安全的，但对于全局变量和共享数据是不安全的，需要进行同步处理。而对于这样多实例的情况SingleThreadModel接口并不能解决并发访问问题。

Servlet体系结构是建立在Java多线程机制之上的，它的生命周期是由Web容器负责的。当客户端第一次请求某个Servlet时，Servlet容器将会根据web.xml配置文件实例化这个Servlet类。当有新的客户端请求该Servlet时，一般不会再实例化该Servlet类，也就是有多个线程在使用这个实例。这样，当两个或多个线程同时访问同一个Servlet时，可能会发生多个线程同时访问同一资源的情况，数据可能会变得不一致。所以在用Servlet构建的Web应用时如果不注意线程安全的问题，会使所写的Servlet程序有难以发现的错误。

实例变量不正确的使用是造成Servlet线程不安全的主要原因。下面针对该问题给出了三种解决方案并对方案的选取给出了一些参考性的建议。

1、实现 SingleThreadModel 接口

该接口指定了系统如何处理对同一个Servlet的调用。如果一个Servlet被这个接口指定,那么在这个Servlet中的service方法将不会有两个线程被同时执行，当然也就不存在线程安全的问题。这种方法只要将前面的Concurrent Test类的类头定义更改为：

```
Public class Concurrent Test extends HttpServlet implements SingleThreadModel {  
    .....  
}
```

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
💬 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

2、同步对共享数据的操作

使用synchronized 关键字能保证一次只有一个线程可以访问被保护的区段，在本论文中的Servlet可以通过同步块操作来保证线程的安全。同步后的代码如下：

```
.....  
Public class Concurrent Test extends HttpServlet { .....  
Username = request.getParameter ("username");  
Synchronized (this){  
Output = response.getWriter ();  
Try {  
Thread. Sleep (5000);  
} Catch (InterruptedException e){  
output.println("用户名:"+Username+"  
");  
}  
}  
}
```

0

3、避免使用实例变量

本实例中的线程安全问题是实例变量造成的，只要在Servlet里面的任何方法里面都不使用实例变量，那么该Servlet就是线程安全的。

修正上面的Servlet代码，将实例变量改为局部变量实现同样的功能，代码如下：

```
.....  
Public class Concurrent Test extends HttpServlet {public void service (HttpServletRequest request, HttpS  
ervletResponse  
Response) throws ServletException, IOException {  
Print Writer output;  
String username;  
Response.setContentType ("text/html; charset=gb2312");  
.....  
}  
}
```

对上面的三种方法进行测试，可以表明用它们都能设计出线程安全的Servlet程序。但是，如果一个Servlet实现了SingleThreadModel接口，Servlet引擎将为每个新的请求创建一个单独的Servlet实例，这将引起大量的系统开销。SingleThreadModel在Servlet2.4中已不再提倡使用；同样如果在程序中使用同步来保护要使用的共享的数据，也会使系统的性能大大下降。这是因为被同步的代码块在同一时刻只能有一个线程执行它，使得其同时处理客户请求的吞吐量降低，而且很多客户处于阻塞状态。另外为保证主存内容和线程的工作内存中的数据的一致性，要频繁地刷新缓存,这也会大大地影响系统的性能。所以在实际的开发中也应避免或最小化 Servlet 中的同步代码；在Servlet中避免使用实例变量是保证Servlet线程安全的最佳选择。从Java 内存模型也可以知道，方法中的临时变量是在栈上分配空间，而且每个线程都有自己私有的栈空间，所以它们不会影响线程的安全。

补充：

Servlet存在的多线程问题

实例变量: 实例变量是在堆中分配的,并被属于该实例的所有线程共享，所以不是线程安全的。

JSP系统提供的8个类变量:

JSP 中 用 到 的 OUT,REQUEST,RESPONSE,SESSION,CONFIG,PAGE,PAGECONXT 是 线 程 安 全 的,APPLICATION在整个系统内被使用,所以不是线程安全的。

局部变量: 局部变量在堆栈中分配,因为每个线程都有它自己的堆栈空间,所以是线程安全的。

静态类:静态类不用被实例化,就可直接使用,也不是线程安全的。

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
💬 QQ客服 💬 客服论坛

关于 招聘 广告服务  百度

©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

外部资源:在程序中可能会有多个线程或进程同时操作同一个资源(如:多个线程或进程同时对一个文件进行写操作).

此时也要注意同步问题. 使它以单线程方式执行,这时, 仍然只有一个实例, 所有客户端的请求以串行方式执行. 这样会降低系统的性能

对于存在线程不安全的类,如何避免出现线程安全问题:

- 1、采用synchronized同步。缺点就是存在堵塞问题。
- 2、使用ThreadLocal(实际上就是一个HashMap),这样不同的线程维护自己的对象，线程之间相互干扰。

ThreadLocal的设计

首先看看ThreadLocal的接口：

```
Object get() ; // 返回当前线程的线程局部变量副本    protected Object  
initialValue(); // 返回该线程局部变量的当前线程的初始值  
void set(Object value); // 设置当前线程的线程局部变量副本的值
```

ThreadLocal有3个方法，其中值得注意的是initialValue()，该方法是一个protected的方法，显然是为了子类重写而特意实现的。该方法返回当前线程在该线程局部变量的初始值，这个方法是一个延迟调用方法，在一个线程第1次调用get()或者set(Object)时才执行，并且仅执行1次。ThreadLocal中的确实实现直接返回一个null：

```
protected Object initialValue() { return null; }
```

ThreadLocal是如何做到为每一个线程维护变量的副本的呢？其实实现的思路很简单，在ThreadLocal类中有一个Map，用于存储每一个线程的变量的副本。比如下面的示例实现：

```
public class ThreadLocal  
{  
    private Map values = Collections.synchronizedMap(new HashMap());  
    public Object get()  
    {  
        Thread curThread = Thread.currentThread();  
        Object o = values.get(curThread);  
        if (o == null && !values.containsKey(curThread))  
        {  
            o = initialValue();  
            values.put(curThread, o);  
        }  
        return o;  
    }  
    public void set(Object newValue)  
    {  
        values.put(Thread.currentThread(), newValue);  
    }  
    public Object initialValue()  
    {  
        return null;  
    }  
}
```

当然，这并不是一个工业强度的实现，但JDK中的ThreadLocal的实现总体思路也类似于此。

ThreadLocal的使用

如果希望线程局部变量初始化其它值，那么需要自己实现ThreadLocal的子类并重写该方法，通常使用一个内部匿名类对ThreadLocal进行子类化，比如下面的例子，SerialNum类为每一个类分配一个序号：

```
public class SerialNum  
{  
    // The next serial number to be assigned  
    private static int nextSerialNum = 0;  
    private static ThreadLocal serialNum = new ThreadLocal()
```

 0





联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

💬 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

```
{
    protected synchronized Object initialValue()
    {
        return new Integer(nextSerialNum++);
    }
};

public static int get()
{
    return ((Integer) (serialNum.get())).intValue();
}
}
```

SerialNum类的使用将非常地简单，因为get()方法是static的，所以在需要获取当前线程的序号时，简单地调用：

int serial = SerialNum.get(); 即可。

在线程是活动的并且ThreadLocal对象是可访问的时，该线程就持有一个到该线程局部变量副本的隐含引用，当该线程运行结束后，该线程拥有的所以线程局部变量的副本都将失效，并等待垃圾收集器收集。

ThreadLocal与其它同步机制的比较

ThreadLocal和其它同步机制相比有什么优势呢？ThreadLocal和其它所有的同步机制都是为了解决多线程中的对同一变量的访问冲突，在普通的同步机制中，是通过对象加锁来实现多个线程对同一变量的安全访问的。这时该变量是多个线程共享的，使用这种同步机制需要很细致地分析在什么时候对变量进行读写，什么时候需要锁定某个对象，什么时候释放该对象的锁等等很多。所有这些都是因为多个线程共享了资源造成的。ThreadLocal就从另一个角度来解决多线程的并发访问，ThreadLocal会为每一个线程维护一个和该线程绑定的变量的副本，从而隔离了多个线程的数据，每一个线程都拥有自己的变量副本，从而也就没有必要对该变量进行同步了。ThreadLocal提供了线程安全的共享对象，在编写多线程代码时，可以把不安全的整个变量封装进ThreadLocal，或者把该对象的特定于线程的状态封装进ThreadLocal。

由于ThreadLocal中可以持有任何类型的对象，所以使用ThreadLocal get当前线程的值是需要进行强制类型转换。但随着新的Java版本（1.5）将模版的引入，新的支持模版参数的ThreadLocal<T>类将从中受益。也可以减少强制类型转换，并将一些错误检查提前到了编译期，将一定程度地简化ThreadLocal的使用。

总结

当然ThreadLocal并不能替代同步机制，两者面向的问题领域不同。同步机制是为了同步多个线程对相同资源的并发访问，是为了多个线程之间进行通信的有效方式；而ThreadLocal是隔离多个线程的数据共享，从根本上就不在多个线程之间共享资源（变量），这样当然不需要对多个线程进行同步了。所以，如果你需要进行多个线程之间进行通信，则使用同步机制；如果需要隔离多个线程之间的共享冲突，可以使用ThreadLocal，这将极大地简化你的程序，使程序更加易读、简洁。

ThreadLocal常见用途：

存放当前session用户

存放一些context变量，比如webwork的ActionContext

存放session，比如spring hibernate orm的session

例子：用 ThreadLocal 实现每线程 Singleton

线程局部变量常被用来描绘有状态“单子”（Singleton）或线程安全的共享对象，或者是通过把不安全的整个变量封装进 ThreadLocal，或者是通过把对象的特定于线程的状态封装进 ThreadLocal。例如，在与数据库有紧密联系的应用程序中，程序的很多方法可能都需要访问数据库。在系统的每个方法中都包含一个 Connection 作为参数是不方便的——用“单子”来访问连接可能是一个虽然更粗糙，但却方便得多的技术。然而，多个线程不能安全地共享一个 JDBC Connection。如清单 3 所示，通过使用“单子”中的 ThreadLocal，我们就能让我们的程序中的任何类容易地获取每线程 Connection 的一个引用。这样，我们可以认为 ThreadLocal 允许我们创建每线程单子。

例：把一个 JDBC 连接存储到一个每线程 Singleton 中

```
public class ConnectionDispenser {
    private static class ThreadLocalConnection extends ThreadLocal {
```



0



联系我们



请扫描二维码联系客服

webmaster@csdn.net

400-660-0108

QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

```
public Object initialValue() {
    return DriverManager.getConnection(ConfigurationSingleton.getDbUrl());
}

private ThreadLocalConnection conn = new ThreadLocalConnection();
public static Connection getConnection() {
    return (Connection) conn.get();
}
}
```

注意：

理论上来说，ThreadLocal是的确是相对于每个线程，每个线程会有自己的ThreadLocal。但已经讲到，一般的应用服务器都会维护一套线程池。因此，不同用户访问，可能会接受到同样的线程变量。此，在做基于ThreadLocal时，需要谨慎，避免出现ThreadLocal变量的缓存，导致其他线程访问到本线程变量。

转载：<http://blog.csdn.net/zitian886/article/details/44361739>

上一篇

Java之CyclicBarrier使用

下一篇

JAVA并发编程学习笔记之CAS操作

评论

servlet是线程安全的吗？

 kingmax54212008 2015年11月10日 22:17 775

前两天，有个人问我“servlet是线程安全的吗？”，我却很难给出一个很具体清晰的回答，今天重新整理一下思路，也复习一下那些被扔回给老师的理论。servlet是线程安全的吗？这个问题，...

Servlet的多线程和线程安全

 qq_24145735 2016年09月04日 17:41 1947

原文地址：纸上得来终觉浅 Servlet的多线程和线程安全线程安全 首先说明一下对线程安全的讨论，哪种情况我们可以称作线程安全？网上对线程安全有很多描述，我比较喜欢《Java并发编程实战》给出的定...

Servlet和JSP的线程安全问题

 treeroot 2004年09月03日 19:11 8409

编写Servlet和JSP的时候,线程安全问题很容易被忽略,如果忽视了这个问题,你的程序就存在潜在的隐患.1.Servlet的生命周期Servlet的生命周期是由Web容器负责的,当客户端第一次请求S...

Servlet—HttpServlet、Servlet线程安全

 Xw_Classmate 2016年02月21日 17:36 662

1. HttpServlet Servlet接口SUN公司定义了两个默认实现类，分别为：GenericServlet、HttpServlet。HttpServlet指能够处理HTTP请求的servle...

Servlet多线程安全问题

 qfs_v 2008年07月15日 09:31 16069

一，servlet容器如何同时处理多个请求。Servlet采用多线程来处理多个请求同时访问，Servlet容器维护了一个线程池来服务请求。线程池实际上是等待执行代码的一组线程叫做工作者线程(Wor...

http://blog.csdn.net/i_will_try/article/details/62215633

9/11

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
💬 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

Jersey (servlet) 的线程安全

u010833547 2016年11月16日 11:21 678

#Jersey的控制器是单例的，因此需要保证多个请求到来时出现多线程并发访问Controller. 处理并发的方法：
- 1.类的访问域的所有成员都采用不可变的（只读）。 - ...



0

Servlet是线程安全的吗？

shenjianxz 2017年02月20日 20:2 2003

今天面试问了这个，我当场猜！！！没去了解这个！！！！，丢人了 总结下！，忘了servlet不能初始化一次！！ Servlet 有实例对象 属性，静态变量就是线程不安全的！！，反正安全！ ...



Servlet如何实现多线程安全？？

junli0310 2007年09月28日 08:50 3501

编写Servlet和JSP的时候,线程安全问题很容易被忽略,如果忽视了这个问题,你的程序就存在潜在的隐患.1.Servlet的生命周期Servlet 的生命周期是由Web容器负责的,当客户端第一次请求...

Servlet的线程安全

yaerfeng 2012年03月23日 09:37 16355

Servlet的多线程机制 1. 变量的线程安全：这里的变量指字段和共享数据（如表单参数值 参数变量本地化。多线程并不共享局部变量，所以要尽可能地在servlet中使用局部变...



Servlet线程安全问题的详解

after_you 2017年01月07日 13:20 397

摘要：介绍了Servlet多线程机制，通过一个实例并结合Java 的内存模型说明引起Servlet线程不安全的原因，给出了保证Servlet线程安全的三种解决方案，并说明三种方案在实际开发中的取舍。...

Servlet的线程安全问题

enterys 2015年10月28日 11:12 1137

我们知道当多个客户端并发访问同一个Servlet时，web服务器会为每一个客户端的访问请求创建一个线程，并在这个线程上调用Servlet的service方法，因此service方法内如果访问了同一个资...

java-web之Servlet中的线程安全问题

一，Servlet容器如何同时处理多个请求。Servlet A1023824314 2016年07月25日 00:01 2871
let采用多线程来处理多个请求同时访问，Servlet容器维护了一个线程池来服务请求。线程池实际上是等待执行代码的一组线程叫做工作者线...

Servlet线程安全问题——ThreadLocal模式

线程安全的概念范畴： 线程安全，指的是在多线程 yx0628 2014年03月14日 10:01 1967
程环境下，一个类在执行某个方法时，对类的内部实例变量的访问是安全的。如果代码所在的进程中有多线程在同时运行，而这些线程可能会同时运行这段代...

Servlet线程安全性问题总结

sinat_28978689 2017年07月08日 09:48 398

Servlet线程安全

联系我们



请扫描二维码联系客服

 webmaster@csdn.net

 400-660-0108

 QQ客服  客服论坛

关于 招聘 广告服务  百度

©1999-2018 CSDN版权所有

京ICP证09002463号



经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

servlet是单例的 所以需要线程安全 以及如何实现线程安全 - CSDN博客

Servlet为什么是线程不安全的？ gjf281 2014年04月21日 18:30  1106



"Servlet的线程安全问题只有在大量的并发访问时才会显现出来，并且很难发现，因此在编写Servlet程序时要特别注意。线程安全问题主要是由实例变量造成的，因此在Servlet中应避免使用实例变量。 ...





0







Servlet是否线程安全 zyl1042635242 2015年08月11日 11:09  1100



servlet是线程安全的吗？这个问题，在网上没有看到一个确切的答案，所以我们来分析一下： 首先什么是线程安全？ 引用概念：如果你的代码所在的进程中有多线程在同时运行，而这些线程可能会同时运行这段...

Servlet其实是单例多线程 xiaojiahao_kevin 2016年06月29日 11:48  8085



Servlet如何处理多个请求访问？ Servlet容器默认是采用单实例多线程的方式处理多个请求的： 1.当web服务器启动的时候（或客户端发送请求到服务器时），Servlet就被加载并实例化(只存在...

Servlet 是否线程安全 看完便知 xiaocainiaoxiao 2016年03月10日 22:07  364

摘要：介绍了Servlet多线程机制，通过一个实例并结合Java 的内存模型说明引起Servlet线程不安全的原因，给出了保证Servlet线程安全的三种解决方案，并说明三种方案在实际开发中的取舍。 ...

servlet是线程安全的吗 u013963737 2016年03月07日 16:02  296

servlet是线程安全的吗？ 首先什么是线程安全？ 引用概念：如果你的代码所在的进程中有多线程在同时运行，而这些线程可能会同时运行这段代码。如果每次运行结果和单线程运行的结果是一样的，而且其他...

【Java之Servlet（二）】servlet是单例多线程，以及多线程下如何保...
Servlet 单例多线程 1、Servlet如何处理多个请求访问？ 答：Servlet容器默认是采用单实例多线程的方式处理多个请求的： 1.当web服务器启动的时候（或客户端发送请求到服务器时） ...
 cangchen 2015年04月22日 16:45  2523