

线程封闭

2018年06月14日 00:13:57

版权声明：我挥舞着键盘和本子
线程封闭的三种方法

- Ad_hoc线程封闭：程序
- 堆栈封闭：局部变量，
- ThreadLocal线程封闭

Java中的ThreadLocal类

用，但是这两个线程依然不能看到彼此的ThreadLocal变量域。



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](#) [网站地图](#)

©2018 CSDN版权所有 京ICP证09002463号

🐶 百度提供搜索支持

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

[_28938627/article/details/80686993](#)

所以不会产生并发问题

操作。因此，尽管有两个线程同时执行一段相同的代码，而且这段代码又有一个指向同一个ThreadLocal变量的引

ThreadLocal的实现原理

在ThreadLocal中有一个map

，map的存储的是线程的id，map的value存储的是使用ThreadLocal存储的对象

在ThreadLocal中有四个方法

它主要由四个方法组成initialValue(), get(), set(T), remove(), 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：

```
1  protected T initialValue() {  
2      return null;  
3  }  
4  
5  public T get() {  
6      Thread t = Thread.currentThread();  
7      ThreadLocalMap map = getMap(t);  
8      if (map != null) {
```

```

9      Thr  联系我们
10      if
11
12      请扫描二维码联系客服
13      ① webmaster@csdn.net
14      ② 400-660-0108
15      ③ QQ客服 ④ 客服论坛
16      }
17      }
18      return
19  }
20  public void Thread
21  ThreadL
22  if (map
23  map
24  else
25  cre  经营性网站备案信息
26  网络110报警服务
27  中国互联网举报中心
28  北京互联网违法和不良信息举报中心
29  }
30  public void remove() {
31      ThreadLocalMap m = getMap(Thread.currentThread());
32      if (m != null)
33          m.remove(this);
34  }

```

0

具体使用

可以使用

```

1  ThreadLocal requestHolder = new ThreadLocal();
2  //创建一个ThreadLocal变量
3  requestHolder.set("abc")
4  String result = (String)requestHolder .get();

```

实例化了一个ThreadLocal对象。每个线程仅需要实例化一次即可。虽然不同的线程执行同一段代码时，访问同一个ThreadLocal变量，但是每个线程只能看到私有的ThreadLocal实例。所以不同的线程在给ThreadLocal对象设置不同的值时，他们也不能看到彼此的修改。

```
1
2 public class Re
3
4     private fin
5
6     public stat
7         request
8     }
9
10    public stat
11        return
12    }
13
14    public stat
15        request
16    }
17 }
18
```

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 网站地图

©2018 CSDN版权所有 京ICP证09002463号

🐶 百度提供搜索支持

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

er = new ThreadLocal<>();

0

人工智能工程师凭什么这么值钱

人工智能技术向前发展，也必然会出现一些岗位被人工智能取代，但我们相信，随着人工智能的发展，会有更多的新的、属于未来的工作岗位出现，是社会发展的必然产物，我们能做的也许只能是与时俱进了



慕课网实战·高并发探索（七）：线程封闭 - ThreadLocal

 238

特别感谢：慕课网jimin老师的《Java并发编程与高并发解决方案》课程，以下知识点多数来自老师的课程内容。jimin老师课程地址：J...

Java并发编程--线程封闭 (Ad-hoc封闭 栈封闭 ThreadLocal)

 56

线程封闭实现好的并发是一件困难的事情，所以很多时候我们都想躲避并发。避免并发最简单的方法就是线程封闭。什么是线程封闭呢...

java并发---线程封闭

线程封闭一般通过以下三个

堆栈的一些理解 ~ ~

堆栈是一种执行“后进先出”算

堆栈错误

写此文的目的是为了说明：

从并发编程到分布式系

面试互联网公司不得不说的



对于程序

不背单词和语法，一个公式学好英语

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 网站地图

©2018 CSDN版权所有 京ICP证09002463号

🐾 百度提供搜索支持

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

见

👤 97

糟糕，忽略 2.堆栈封闭：局部变量，无并发问题 3.Thr...

🔄 723

一端封闭的竹筒。有若干个写有编号的小球，小球的直径...

👤 1598

”，堆栈错误因为可能影响安全，所以在异常处理时被加以...

🌐 91

思路，也会聊聊自己遇到的那些坑，以此为记，希望鞭策...

关于堆栈的讲解(我见过的最经典的)

🔄 1.9万

这是一篇转发的文章，我对他进行了格式化而已，原文出处不详。一、预备知识—程序的内存分配一个由c/C++编译的程序占用的内存...

java并发之线程封闭(二)

👤 6610

线程封闭 实现好的并发是一件困难的事情，所以很多时候我们都想躲避并发。避免并发最简单的方法就是线程封闭。什么是线程封闭...

聊聊高并发结合实例说说线程封闭和背后的设计思想

👤 1032

高并发问题抛去架构层面的问题，落实到代码层面就是多线程的问题。多线程的问题主要是线程安全的问题（其他还有活跃性问题，性...

Java多线程之线程封闭(三)

👤 326

线程封闭的概念访问共享变量时，通常要使用同步，所以避免使用同步的方法就是减少共享数据的使用，这种技术就是线程封闭。实现...

相关热词

线程和线程 线程线程 线程在 线程与this 线程】

个人资料



守拙的胖胖

原创

16

粉丝

0

喜欢

1

等级:

博客 2

积分:

187

勋章:



访问:

排名:

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

网站地图

©2018 CSDN版权所有 京ICP证09002463号

 百度提供搜索支持

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

最新文章

ReentranLock 和synchronized区别

AQS之CyclicBarrier

AQS之Semaphore

Java线程池

java 多线程编程

0

https://blog.csdn.net/qq_28938627/article/details/80686993

5/6

个人分类

android

web前端

java

集群

归档

2018年7月

2018年6月

2018年5月

2018年4月

2016年9月

展开

热门文章

chronoment暂停方法的实现
阅读量：341

Nginx+Tomcat搭建集群
阅读量：260

安卓网络下载图片picasso的使用
阅读量：241

ReentranLock 和synchronized区别
阅读量：178

项目中读取配置文件的两种方式
阅读量：172

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

网站地图

©2018 CSDN版权所有 京ICP证09002463号

 百度提供搜索支持

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

0

https://blog.csdn.net/qq_28938627/article/details/80686993

6/6