



## threadlocal原理及常用应用场景

osc\_n0mj0tg4 2018/01/08 14:18 阅读数 39

昇腾众智计划火热上线！140个算子/模型等你来挑战！>>>[HOT](#)

### 1.深入解析ThreadLocal类

ThreadLocal类提供的几个方法：

- public T get() {}
- public void set(T value) {}
- public void remove() {}
- protected T initialValue() {}

get()方法是用来获取ThreadLocal在当前线程中保存的变量副本，set()用来设置当前线程中变量的副本，remove()用来移除当前线程中变量的副本，initialValue()是一个protected方法，一般是用来在使用时进行重写的，它是一个延迟加载方法

一个以ThreadLocal对象为键、任意对象为值的存储结构。

是一个数据结构，有点像HashMap，可以保存"key : value"键值对，但是一个ThreadLocal只能保存一个，并且各个线程的数据互不干扰。

该结构被附带在线程上，也就是说一个线程可以根据一个ThreadLocal对象查询到绑定在这个线程上的一个值。

```
ThreadLocal<String> local = new ThreadLocal();
local.set("test");
String demo = local.get();
```

在线程A中初始化了一个ThreadLocal对象local，并set了一个值test，同时在线程A中通过get可拿到之前设置的值，但是如果在线程B中，拿到的将是一个null

这是如何实现的呢？之前说过，ThreadLocal保证了各个线程的数据互不干扰  
看看set(T value)和get()方法的源码

```
// 返回当前线程该线程局部变量副本中的值
public T get() {
    Thread t = Thread.currentThread();
    ThreadLocalMap map = getMap(t);
    if (map != null) {
        ThreadLocalMap.Entry e = map.getEntry(this);
        if (e != null) {
            @SuppressWarnings("unchecked")
            T result = (T)e.value;
            return result;
        }
    }
    return setInitialValue();
}
/**
 *设置此线程局部变量的当前线程的副本到指定的值
 *大多数的子类都不需要重写此方法
 */
public void set(T value) {
    Thread t = Thread.currentThread();
    ThreadLocalMap map = getMap(t);
    if (map != null)
        map.set(this, value);
    else
        createMap(t, value);}

/**
 * Get the map associated with a ThreadLocal. Overridden in
 * InheritableThreadLocal.
 *
 * @param t the current thread
 * @return the map
 */
ThreadLocalMap getMap(Thread t) {
    return t.threadLocals;
}
```

可以发现，每个线程中都有一个ThreadLocalMap数据结构

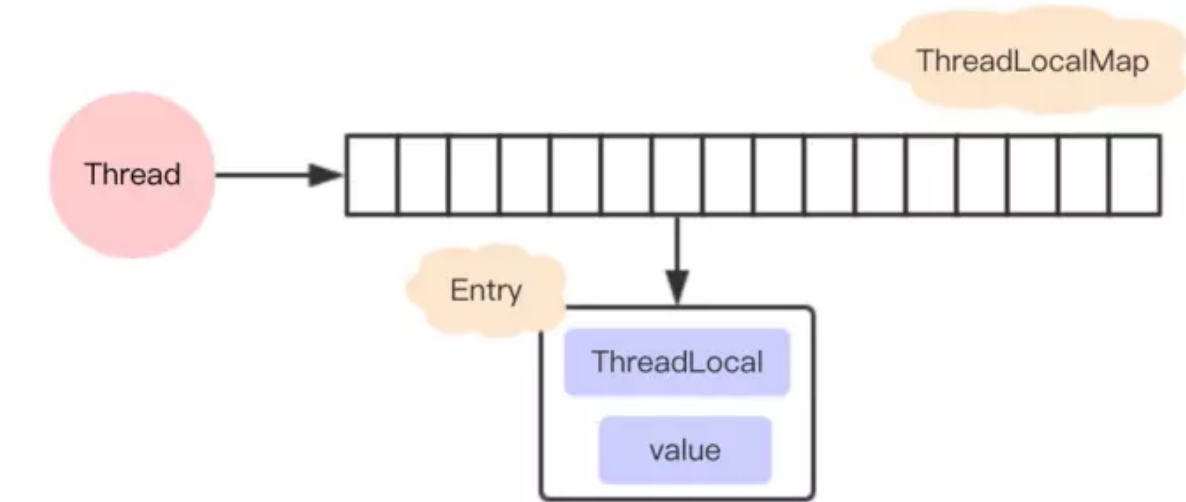
执行set时，其值是保存在当前线程的threadLocals变量中，执行get时，从当前线程的threadLocals变量获取

所以在线程A中set的值，对线程B来说是摸不到的，而且在线程B中重新set的话，也不会影响到线程A中的值，保证了线程之间不会相互干扰

ThreadLocalMap是一个类似HashMap的数据结构，但是在ThreadLocal中，并没实现Map接口;

在ThreadLoalMap中，也是初始化一个大小为16的Entry数组table，Entry节点对象用来保存每一个key-value键值对，这里的key永远都是ThreadLocal对象.通过ThreadLocal对象的set方法，结果把ThreadLocal对象自己当做key;

ThreadLoalMap的Entry是继承WeakReference，和HashMap很大的区别是，Entry中没有next字段，所以不存在链表的情况;



### 2.需要注意：

关于作者

**osc\_n0mj0tg4**  
[♡ 关注](#) [✉ 私信](#) [💬 提问](#)

文章	经验值	粉丝	关注
500	0	0	0

源创计划 [立即入驻](#)

自媒体入驻开源社区，  
获百万流量，打造个人技术品牌

推荐关注 [换一批](#)

- 太原随风**  
文章 2 访问 449
- 开源中国首席辣条代理**  
文章 4 访问 5K
- inkind**  
文章 12 访问 2.3W
- 大王叫下**  
文章 27 访问 3.2W
- 雾海树妖**  
开源软件作者



打赏



0 评论



0 收藏



0 赞



分享

Gitee

Linux 进阶教程，从基础到高级，一步步教你掌握 Linux 系统管理知识

另外，说ThreadLocal使得各线程能够保持各自独立的一个对象，并不是通过set()来实现的，而是通过每个线程中的new 对象 的操作来创建的对象，每个线程创建一个，不是什么对象的拷贝或副本。

通过set()将这个新创建的对象引用保存到各线程的自己的一个map中，每个线程都有这样一个map，执行get()时，各线程从自己的map中取出放进去的对象，因此取出来的是各自自己线程中的对象，ThreadLocal实例是作为map的key来使用的。

本文转载自：<https://www.cnblogs.com/xushuai0620/p/8242458.html>

🚩 举报



打赏



0 赞



0 收藏



分享

作者的其它热门文章

[atomikos + druid 连接超时失效](#)

[H265 Profile & Level & Tier 介绍](#)

[apache 运行一段时间出现错误](#)

[echarts 折线图点击高亮](#)



其他人还在看

[更多精彩内容](#)

### 一文搞定ThreadLocal原理

ThreadLocal是什么 ThreadLocal是一个关于创建线程局部变量的类。通常情况下，我们创建的变量是可以被任何一个线程访问并修改的。而使用ThreadLocal创建的变量只能被当前线程访问，其他线程则无法访问和修改。 Th...

💬 0 ⭐ 0

### JAVA并发编程阶段总结篇，解决死锁问题以及ThreadLocal原理分析

一、线程的死锁问题 首先来看下死锁的概念吧：一组相互竞争的线程因为相互等待，造成“永久”阻塞的现象，我们称之为死锁；那么有死锁必然就有活锁了，什么是活锁呢？即任务或者执行者都没有被阻塞，由于某些条件...

💬 0 ⭐ 0

### 美团面试问我ThreadLocal原理，我这么回答通过了面试！

上周我侥幸通过美团一面，岗位是Java后端开发工程师。美团面试官给我进行了二面。面试过程中他问了ThreadLocal原理（上次问线程池，这次问ThreadLocal，美团爸爸这么喜欢线程安全机制么），今天详细讲一讲Threa...

💬 0 ⭐ 0

### 简析ThreadLocal原理及应用

简析ThreadLocal原理及应用 原创： 东晨雨 JAVA万维猿圈 4月17日 ThreadLocal的源码加上注释不超过八百行，源码结构清晰，代码也比较简洁。ThreadLocal可以说是Java中解决多线程数据共享问题方案中的一股清流，该方案为每个线程分配一个独立的变量副本， ...

💬 0 ⭐ 0

### 007-多线程-基础-ThreadLocal原理分析-线程变量副本

一、简介 早在JDK 1.2的版本中就提供java.lang.ThreadLocal，ThreadLocal为解决多线程程序的并发问题提供了一种新的思路。使用这个工具类可以很简洁地编写出优美的多线程程序。当使用ThreadLocal维护变量时，ThreadLocal为每个使用该变量的线程提供...

💬 0 ⭐ 0

### ThreadLocal原理分析及内存泄漏代码展示

今日福利：扫描上方二维码，回复【黑马】黑马Java Springboot2(1).0开发头条项目（资料、代码、讲义）14天完整版 背景知识 引用 Object o = new Object(); 这个o，我们可以称之为对象引用，而new Object()我们可...

💬 0 ⭐ 0

### 增强版的ThreadLocal-TransmittableThreadLocal原理概述

一、前言 前面讲解了增强版的ThreadLocal-TransmittableThreadLocal可以优雅解决线程变量的继承问题，本节我们来探讨其实现。二、TransmittableThreadLocal实现原理 简单回顾如何使用TransmittableThreadLocal解...

💬 0 ⭐ 0

### 深度解析ThreadLocal原理

今天呢，和大家聊一下ThreadLocal。 1. 是什么？ JDK1.2提供的的一个线程绑定变量的类。 他的思想就是：给每一个使用到这个资源的线程都克隆一份，实现了不同线程使用不同的资源，且该资源之间相互独立 2. 为什么...

💬 2 ⭐ 8

### 浅谈ThreadLocal原理

前言： 经过一宿的面壁思过，下文写的对ThreadLocal的源码分析，过于枯燥乏味，现在简单的总结一下ThreadLocal，提高一下做人的level。 ThreadLocal的作用就是：线程安全。 ThreadLocal的本质就是一个内部的静态的...

💬 0 ⭐ 1

### ThreadLocal原理探究

2.1.11 ThreadLocal 多线程访问同一个共享变量特别容易出现并发问题，特别是多个线程需要对一个共享变量进行写入时候，为了保证线程安全，一般需要使用者在访问共享变量的时候进行适当的同步，如下图： 同步的措施...

💬 0 ⭐ 0

¥

打赏

💬

0 评论

★

0 收藏

👍

0 赞

➦

分享

#### OSCHINA 社区

关于我们  
联系我们  
加入我们  
合作伙伴  
Open API

#### 在线工具

Gitee.com  
企业研发管理  
CopyCat-代码克隆检测  
实用在线工具  
**国家反诈中心APP下载**

#### QQ交流群



895991038

#### 微信公众号



聚合全网技术文章，根据你的阅读喜好进行个性推荐

下载 APP

¥

打赏

💬

0 评论

★

0 收藏

👍

0 赞

➦

分享