链滴

java 中 ThreadLocal 的简单理解

作者：umeone

原文链接：https://ld246.com/article/1515658125837

来源网站：链滴

ThreadLocal

● 从JDK1.2版本开始提供了ThreadLocal类，以一种新的思路解决多线程并发问题。使用ThreadLoca实例维护多线程变量时，不会出现线程安全问题。虽然多个线程共用一个ThreadLocal实例，但在变的使用上，各线程之间对变量不可见。

ThreadLocal的使用

```
ThreadLocal threadLocal = new ThreadLocal();
```

● ThreadLocal提供无参的构造函数进行实例创建。在使用方面，该类也仅提供了几个简单的方法进操作：

1. set方法：　threadLocal.set("threadLocal test")对ThreadLocal进行值得设置。
2. get方法：　Object test = threadLocal.get()进行值得获取。
3. remove方法：　threadLocal.remove();删除对应线程中存储的值。

● 在JDK1.5版本开始，ThreadLocal提供泛型的支持，在获取值时不再需要需要进行类型强转。

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

源码浅析

● 通过构造函数我们发现，ThreadLocal在构造函数中并没有做任何操作，所以我们从set方法作为入进行分析。

```
public void set(T value) {
  Thread t = Thread.currentThread();
  ThreadLocalMap map = getMap(t);
 if (map != null)
    map.set(this, value);
 else
   createMap(t, value);
}
```

通过set方法我们可以看到，ThreadLocal实际是在其内部维护了一个ThreadLocalMap实例进行变的存储，并把该实例与线程进行绑定。在ThreadLocalMap中key的值为this，也就是当前线程的ThradLocal实例，value为当前设置的值

● 在get方法中，通过调用　ThreadLocalMap map = getMap(t);获取当前线程绑定的ThreadLocal ap实例，并获取value值。

● 值得注意的是ThreadLocal同一线程只能存储一个值，如果需要存储多个值，可以考虑封装成一个象进行存储。

默认值

原文链接：java 中 ThreadLocal 的简单理解

● ThreadLocal通过 protected T initialValue()方法进行value的初始化。

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

● 通过源码我们可以看到，该方法直接返回 null。所以在创建实例后直接调用get方法时，我们得到是空值。如果我们需要在ThreadLocal实例创建的时候进行初始化，可以通过继承ThreadLocal并重写nitialValue()方法即可。

● 那么 initialValue()方法在何时进行调用的呢？通过查找源码我们在get方法中看到了该方法的调。

```
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();
```

● 通过get方法执行流程可以看到，当 **ThreadLocal**创建实例的时候，ThreadLocalMap map = ge Map(t);返回的map值肯定为空，所以执行setInitialValue();并返回，我们进入该方法：

```
private T setInitialValue() {
  T value = initialValue(); //目标代码
  Thread t = Thread.currentThread();
  ThreadLocalMap map = getMap(t);
 if (map != null)
    map.set(this, value);
 else  createMap(t, value);
   return value;
}
```

● 从setInitialValue方法实现可以看到，在该方法中调用了initialValue返回了我们自定义的返回值，设置到ThreadLocalMap实例中。

● 通过上面的查看源码可以发现，ThreadLocal在设置初始化值时，需要在第一次调用get方法时才进行初始化，而不是在实例创建的时候进行初始化。

以上即为个人对ThreadLocal的简单理解。