

原 慕课网实战·高并发探索（七）：线程封闭 - ThreadLocal

2018年04月11日 20:50:16

版权声明：本文为博主原创文章，未经博主允许不得转载。

特别感谢：慕课网jimin老师

jimin老师课程地址：Java并发编程

1、什么是线程封闭

它其实就是把对象封装到ThreadLocal中，那么这个对象就算不是线程安全的，也不会出现任何线程安全方面的问题。

线程封闭技术有一个常见的应用：

数据库连接对应jdbc的Connection对象，Connection对象在实现的时候并没有对线程安全做太多的处理，jdbc的规范里也没有要求Connection对象必须是线程安全的。

实际在服务器应用程序中，线程从连接池获取了一个Connection对象，使用完再把Connection对象返回给连接池，由于大多数请求都是由单线程采用同步的方式来处理的，并且在Connection对象返回之前，连接池不会将它分配给其他线程。因此这种连接管理模式处理请求时隐含的将Connection对象封闭在线程里面，这样我们使用的connection对象虽然本身不是线程安全的，但是它通过线程封闭也做到了线程安全。

2、线程封闭的种类：

(1) Ad-hoc 线程封闭：

Ad-hoc线程封闭是指，维护线程封闭性的职责完全由程序实现来承担。Ad-hoc线程封闭是非常脆弱的，因为没有任何一种语言特性，例如可见性修饰符或局部变量，能将对象封闭到目标线程上。事实上，对线程封闭对象（例如，GUI应用程序中的可视化组件或数据模型等）的引用通常保存在公有变量中。

(2) 堆栈封闭：

堆栈封闭其实就是方法中定义局部变量。不存在并发问题。多个线程访问一个方法的时候，方法中的局部变量都会被拷贝一份到线程的栈中（Java内存模型），所以局部变量是会被多个线程所共享的。



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

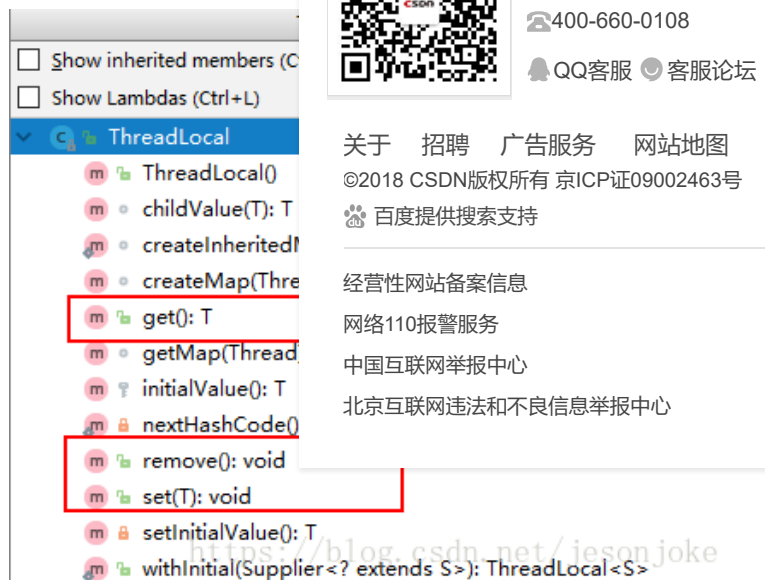
ThreadLocal

om。 <https://blog.csdn.net/jesonjoke/article/details/79902773>

课程，以下知识点多数来自老师的课程内容。

象，那么这个对象就算不是线程安全的，也不会出现任何线程安全方面的问题。

(3) ThreadLocal线程变量
它是一个特别好的封闭变量，提供了get、set、remove方法，每个操作都是线程安全的。



map, map的key是每个线程的名称，而map的value就是我们要封闭的变量。

ThreadLocal提供了get、set、remove方法。

```

1 //ThreadLocal的get方法源码
2 public T get() {
3     Thread t = Thread.currentThread(); //当前线程对象
4     ThreadLocalMap map = getMap(t); //get操作基于当前线程
5     if (map != null) {
6         ThreadLocalMap.Entry e = map.getEntry(this);
7         if (e != null) {
8             @SuppressWarnings("unchecked")
9             T result = (T)e.value;
10            return result;
11        }
12    }
13    return setInitialValue();
14 }

```

3、Springboot框架

说明：

- 1、这里不描述springboot
- 2、我们这里的例子使用

coding：

(1) 创建一个包含ThreadLocal

```
1 public class RequestHolder {
2
3     private final ThreadLocal<Request> requestHolder =
4         new ThreadLocal<>();
5
6     public static Request request() {
7         return requestHolder.get();
8     }
9
10    public static Long getId() {
11        return requestHolder.get().getId();
12    }
13
14    public static void remove() {
15        requestHolder.remove();
16    }
17 }
```

(2) 创建Filter，在Filter中对ThreadLocal做添加操作。

```
1 public class HttpFilter implements Filter {
2
3     @Override
4     public void init(FilterConfig filterConfig) throws ServletException {
5
6     }
7
8     @Override
```

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

以正常运行的springboot简单项目。

不使用threadLocal，对于Springboot的filter与Interceptor不做过多的

余、获取操作。

```
er = new ThreadLocal<>();
```

```

9     public void
10
11
12     HttpSer
13     //打印当
14     log.inf
15     //将当前
16     Request
17     filterC
18 }
19
20 @Override
21 public void
22
23 }
24 }

```

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

```

,
se,
ws IOException, ServletException {
st) servletRequest;

read().getId(), request.getServletPath());

());
sponse);

```

1

(3)创建controller，在controller中获取到filter中存入的值

```

1 @Controller
2 @RequestMapping("/threadLocal")
3 public class ThreadLocalController {
4
5     @RequestMapping("/test")
6     @ResponseBody
7     public Long test() {
8         return RequestHolder.getId();
9     }
10 }

```

(4) 创建拦截器Interceptor，在拦截器中删除刚才添加的值

```

1 public class HttpInterceptor extends HandlerInterceptorAdapter {
2
3     @Override
4     public boolean preHandle(HttpServletRequest request, HttpServletResponse response,
5                             Object handler) throws Exception {

```

```

6      log.info("...");
7      return response;
8  }
9
10     @Override
11     public void doGet(
12         HttpServletRequest request,
13         HttpServletResponse response,
14         long timeout)
15         throws IOException,
16         ServletException {
17     }

```

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

```

request, HttpServletResponse response,
long timeout) throws Exception {

```

(5) 在springboot的启动类中

继承WebMvcConfigurerAdapter 类。（我这里的启动类名为：ConcurrencyApplication）

```

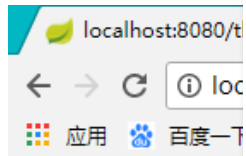
1  @SpringBootApplication
2  public class ConcurrencyApplication extends WebMvcConfigurerAdapter {
3
4      public static void main(String[] args) {
5          SpringApplication.run(ConcurrencyApplication.class, args);
6      }
7
8      //注册过滤器
9      @Bean
10     public FilterRegistrationBean httpFilter() {
11         FilterRegistrationBean registrationBean = new FilterRegistrationBean();
12         registrationBean.setFilter(new HttpFilter());
13         //设置要过滤的url
14         registrationBean.addUrlPatterns("/threadLocal/*");
15         return registrationBean;
16     }
17
18     //注册拦截器
19     @Override
20     public void addInterceptors(InterceptorRegistry registry) {
21         registry.addInterceptor(new HttpInterceptor()).addPathPatterns("/**");
22     }

```

```
23     }  
    }
```

(6) 运行程序，访问 [http://localhost:8080/threadLocal/test](#)

- 页面中打印出我们当前线程ID为40



40

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

下：

- 查看控制台：

```
com.sun.net.httpserver.HttpHandler : do filter, 40, /threadLocal/test  
com.sun.net.httpserver.HttpHandler : preHandle  
com.sun.net.httpserver.HttpHandler : afterCompletion
```

从控制台的打印日志我们可以看出，首先filter过滤器先获取到我们当前的线程ID为40、我们当前的请求路径为/threadLocal/test，紧接着进入了我们的Interceptor的preHandle方法中，打印了preHandle字样。最后进入了我们的Interceptor的afterCompletion方法，删除了我们之前存入的值，并打印了afterCompletion字样。

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，以上知识点多数来自老师的课程内容。

jimin老师课程地址：[Java并发编程与高并发解决方案](#)

人工智能工程师凭什

人工智能技术向前发展，也
能是与时俱进了



想对作者说点什么

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

线程封闭

线程封闭的三种方法 Ad_ho

Java并发编程--线程挂

线程封闭实现好的并发是一

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

利用ThreadLocal管理request和session以及用户信息，实现 Use anywhere

1.我们有时需要获取request或session中的数据的时候，首先需要获取request和session对象，这个通常是在Controller的时候当做入参...

并发编程与高并发解决方案学习(线程封闭-ThreadLocal)

一、概念线程封闭就是把对象封装到一个线程里，只有一个线程能看到这个对象，那么这个对象不是线程安全的，也不会出现线程安全...

java并发之线程封闭(二)

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...

慕课网实战·高并发探索（九）：并发容器 J.U.C - 线程安全的集合与Map

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...



码农怎能不懂英语？！试试这个数学公式

老司机教你一个数学公式秒懂天下英语

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...展的必然产物，我们能做的也许只

1



35

封闭：局部变量，Java线程访问局部变量时，会拷贝一份到...



56

al)

避免并发最简单的方法就是线程封闭。什么是线程封闭呢...



644



86



6610



268

聊聊**高并发**（二）总结

高并发问题抛去架构层面的问题，主要是线程安全的问题（其他还有活跃性问题，性能问题主要是线程安全的问题（其他还有活跃性问题，性能...

慕课网**实战·高并发探索**

特别感谢：慕课网jimin老师

Java**并发编程**：**线程封闭**

转载请标明出处：http://blog.csdn.net/jesonjoke

慕课网**实战·高并发探索**

特别感谢：慕课网jimin老师

相关热词

慕课网nodejs

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

👤 👁 8053

的问题主要是线程安全的问题（其他还有活跃性问题，性能...

👤 👁 489

下知识点多数来自老师的课程内容。jimin老师课程地址：J...

🌐 👁 3万

出自方志朋的博客 什么是线程封闭当访问共享变量时，往...

👤 👁 261

下知识点多数来自老师的课程内容。jimin老师课程地址：J...

👤 👁 1606

把之前版本的稍作修改就可以更加方便的使用分页了。只是把之前版本的page对象放到了threadlocal中了。取page也从threadlocal中...

👤 👁 1736

Spring如何处理**线程**并发问题？Spring使用ThreadLocal解决线程安全问题 我们知道在一般情况下，只有无状态的Bean才可以在多线程...

👤 👁 2436

线程封闭（Thread Confinement）：仅在单线程内访问数据，不需要同步。常见应用是：JDBC（Java Database Connectivity）的Co...

🔄 👁 1192

利用**threadLocal** 把拦截器中的对象传递到controller或服务中

可以用 request 携带数据。更优雅的方式是用threadlocal。请求进入tomcat 和产生响应前，都处于同一个线程中比如在一个登录拦截器...

🔔 👁 1935

ThreadLocal应用之一-----利用拦截器或过滤器设置请求上下文

之前一直使用ThreadLocal进行请求上下文的设置，只知道ThreadLocal是线程安全的每个线程获取的是本线程所对应的值，没有去深入...

1

https://blog.csdn.net/jesonjoke/article/details/79902773

8/11

threadLocal

1、 ThreadLocal存放的值是

java ThreadLocal及线

当访问共享的可变数据时，

java线程封闭 Thread

当访问共享的可变数据时

2018慕课网Java秒杀

第1章 课程介绍及项目框架

慕课网-java高并发秒杀api之高并发优化-总结

1.架构 2.事务，锁，网络延迟， spl procedure 3.集群化部署， nginx负载均衡

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

👤 359

一些数据，避免通过参数来传递 2、 线程的角度看，每个...

👤 156

就可以避免使用同步机制。1.栈封闭java中基本类型的局...

👤 25

就是不共享数据。Java提供了ThreadLocal类来方便开发者...

👤 614

ot工程框架的搭建，集成Thymeleaf，集成Mybatis，安装R...

👤 2314

个人资料



Jeff Osmond

关注

原创	粉丝	喜欢	评论
62	32	46	21

等级: 博客 4 访问: 3万+

积分: 1021 排名: 5万+

勋章:

最新文章

MySQL 各种恶心的统计功能(持续更新中)

Maven使用经验：IDEA 脚手

慕课网实战·高并发探索（十五）：线程池与锁

慕课网实战·高并发探索（十四）：线程池Executor

慕课网实战·高并发探索（十三）：并发容器J.U.C -- 组件FutureTask、ForkJoin、BlockingQueue

博主专栏

新手快速入门-Elasticsearch
阅读量：6156 6 篇

技术探索 - 慕课网实战·高并发探索
阅读量：4398 15 篇

新手快速入门-Activiti工作流
阅读量：0 0 篇

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

归档

2018年9月

2018年6月

2018年5月

2018年4月

2018年3月

展开

热门文章

Servlet学习:(一)使用IDEA搭建Servlet程序

阅读量：5745

linux服务器上配置多个svn仓库

阅读量：3689

ElasticSearch学习（2）--使用Kibana连接ES（5.0以上版本）

阅读量：3507

基于阿里云的Jenkins+Maven+SVN+Tomcat的持续集成（三）-jenkins集成

阅读量：3022

Jersey项目报错：Unrecognized token 'id': was expecting ('true', 'false' or 'null') ...

阅读量：2164

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

1