链滴

Jetty 源码分析

作者：james

原文链接：https://ld246.com/article/1516880944105

来源网站：链滴

1. Jetty线程池

- 参考： jetty源码分析：QueuedThreadPool
- Server类中定义了线程池：

```
类org.eclipse.jetty.server.Server中的属性：

private final ThreadPool _threadPool; //线程池
_threadPool=pool!=null?pool:new QueuedThreadPool(); //默认是一个QueuedThreadPool

private final List<Connector> _connectors = new CopyOnWriteArrayList<>();

类org.eclipse.jetty.util.thread.QueuedThreadPool中的属性：
private final ConcurrentLinkedQueue<Thread> _threads = new ConcurrentLinkedQueue<>();
//线程池以ConcurrentLinkedQueue来容纳，默认maxThreads:200,minThreads:8,idleTimeout:60
00(60s)
private final BlockingQueue<Runnable> _jobs; //任务队列
//jobs队列是如下定义的：queue=new BlockingArrayQueue<>(_minThreads, _minThreads),
创建QueuedThreadPool的时候就初始化了_threads和_jobs的Queue。最小容量是最小线程书，以
值扩展，默认最大容量是Integer.MAX_VALUE

 public Server(@Name("threadpool") ThreadPool pool)
    {
        _threadPool=pool!=null?pool:new QueuedThreadPool();
        addBean(_threadPool);
        setServer(this);
    }


public QueuedThreadPool()
    {
        this(200);
    }

    public QueuedThreadPool(@Name("maxThreads") int maxThreads)
    {
        this(maxThreads, 8);
    }

    public QueuedThreadPool(@Name("maxThreads") int maxThreads,  @Name("minThreads")
nt minThreads)
    { //默认线程超时时间是1分钟
        this(maxThreads, minThreads, 60000);
    }

    public QueuedThreadPool(@Name("maxThreads") int maxThreads,  @Name("minThreads")
nt minThreads, @Name("idleTimeout")int idleTimeout)
    {
        this(maxThreads, minThreads, idleTimeout, null);
    }

    public QueuedThreadPool(@Name("maxThreads") int maxThreads, @Name("minThreads")
nt minThreads, @Name("idleTimeout") int idleTimeout, @Name("queue") BlockingQueue<Ru
nable> queue)
    {
        setMinThreads(minThreads);
        setMaxThreads(maxThreads);
        setIdleTimeout(idleTimeout);
        setStopTimeout(5000);

        if (queue==null)
            queue=new BlockingArrayQueue<>(_minThreads, _minThreads);
        _jobs=queue;

    }

    /**
     * Set the maximum thread idle time.
     * Threads that are idle for longer than this period may be
     * stopped.
     * Delegated to the named or anonymous Pool.
     *
     * @param idleTimeout Max idle time in ms.
     * @see #getIdleTimeout
     */
    public void setIdleTimeout(int idleTimeout)
    {
        _idleTimeout = idleTimeout;
    }

     /**
     * <p>Sets the maximum Idle time for a connection, which roughly translates to the {@link
Socket#setSoTimeout(int)}
     * call, although with NIO implementations other mechanisms may be used to implement t
e timeout.</p>
     * <p>The max idle time is applied:</p>
     * <ul>
     * <li>When waiting for a new message to be received on a connection</li>
     * <li>When waiting for a new message to be sent on a connection</li>
     * </ul>
     * <p>This value is interpreted as the maximum time between some progress being made
n the connection.
     * So if a single byte is read or written, then the timeout is reset.</p>
     *
     * @param idleTimeout the idle timeout
     */
    public void setIdleTimeout(long idleTimeout)
    {
        _idleTimeout = idleTimeout;
    }
```

- jetty中有两个超时时间：

  1. 线程的超时时间默认1分钟，线程运行超过1分钟就会被停止
  2. 连接超时时间默认30秒，底层是socket的超时时间，超时30秒，连接会被关闭。

原文链接：Jetty 源码分析