

```
{
  "2":{"1","3":"AkOg-1637661618616","4":{"fv":"","5":{"3":"55fcen1637585869632","4":{
    "li":"list8","ll":1,"lt":"unordered"},"5":{"2":"2","3":"oTKi-1637661618618","7":{"8":"线程的进程创建
    销毁会使用大量的资源，比如并发情况下的线程，会对性能影响很大"}}},"6":"l"},
    {"3":"93affr1637585913606","4":{"li":"list8","ll":1,"lt":"unordered"},"5":{"2":"2","3":"vP3t-
    1637661618619","7":{"8":"提前创建好多个线程，放入线程池，使用时直接获取，使用完放回线程
    池"}}},"6":"l"},
    {"3":"11xjtb1637585927285","4":{"li":"list8","ll":1,"lt":"unordered"},"5":{
      [{"2":"2","3":"JGG3-1637661618619","7":{"8":"好处"}}],"6":"l"},
      {"3":"48gnxj1637585955940","4":{
        {"li":"list8","ll":2,"lt":"unordered"},"5":{"2":"2","3":"OBCM-1637661618619","7":{"8":"提高了响应速
        度（减少了创建新线程的时间）"}}},"6":"l"},
        {"3":"64towu1637585986086","4":{
          {"li":"list8","ll":2,"lt":"unordered"},"5":{"2":"2","3":"wQLG-1637661618619","7":{"8":"降低了资源消
          耗（重复利用线程池中的资源，不需要每次都创建）"}}},"6":"l"},
          {"3":"15ajqa1637585995204","4":{
            {"li":"list8","ll":2,"lt":"unordered"},"5":{"2":"2","3":"NARZ-1637661618620","7":{"8":"便于线程管
            理"}}},"6":"l"},
            {"3":"36lufd1637586018314","4":{"li":"list8","ll":3,"lt":"unordered"},"5":{
              [{"2":"2","3":"U6nR-1637661618620","7":{"8":"corePoolSize：线程池大小"}}],"6":"l"},
              {"3":"51ledz1637586054852","4":{"li":"list8","ll":3,"lt":"unordered"},"5":{"2":"2","3":"YiEh-
              1637661618620","7":{"8":"maxNumpoolSize：最大线程数量"}}},"6":"l"},
              {"3":"67eukj1637627220472","4":{"li":"list8","ll":3,"lt":"unordered"},"5":{"2":"2","3":"FUP5-
              1637661618620","7":{"8":"keepAliveTime：线程池没有任务时最多保持多长时间会销毁"}}},"6":"l"},
              {"3":"20tevj1637627299078","4":{"li":"list8","ll":1,"lt":"unordered"},"5":{"2":"2","3":"q2xx-
              1637661618620","7":{"8":"线程池相关API（JDK1.5）：ExecutorService和Executors"}}},"6":"l"},
              {"3":"29ipdm1637628197287","4":{"li":"list8","ll":2,"lt":"unordered"},"5":{"2":"2","3":"6mrS-
              1637661618621","7":{"8":"Executors：工具类，线程池的工厂类，用于创建并返回不同类型的线程
              池"}}},"6":"l"},
              {"3":"34rhlx1637586027948","4":{"li":"list8","ll":3,"lt":"unordered"},"5":{
                [{"2":"2","3":"b7ai-1637661618621","7":{"8":"ExecutorService newFixedThreadPool(int nThread)
                创建线程池，池中最多有多少个线程，这个nThread到底是用来干什么的？只知道不能等于0"}}],"6":"l"},
                {"3":"92aosy1637627377298","4":{"li":"list8","ll":2,"lt":"unordered"},"5":{"2":"2","3":"sRc5-
                1637661618620","7":{"8":"ExecutorService：真正的线程池接口，常见子类
                ThreadPoolExecutor"}}},"6":"l"},
                {"3":"13ufcc1637627525178","4":{
                  {"li":"list8","ll":3,"lt":"unordered"},"5":{"2":"2","3":"S3jS-1637661618621","7":{"8":"void
                  execute(Runnable r) 执行任务/命令，没有返回值，一般用来执行Runnable"}}},"6":"l"},
                  {"3":"94bszf1637627630743","4":{"li":"list8","ll":3,"lt":"unordered"},"5":{"2":"2","3":"q4SF-
                  1637661618621","7":{"8":"Future\u003cT\u003e submit(Callable\u003cT\u003e c) 执行任务/命
                  令，有返回值，一般用来执行Callable"}}},"6":"l"},
                  {"3":"92uerz1637627663117","4":{
                    {"li":"list8","ll":3,"lt":"unordered"},"5":{"2":"2","3":"vE8S-1637661618621","7":{"8":"void
                    shutdown() 关闭线程池"}}},"6":"l"},
                    "compress":true}
                }
              }
            }
          }
        }
      }
    }
  }
}
```