

在不同的线程中使用相同的Jedis实例会发生奇怪的错误。但是创建太多的实现也不好因为这意味着会建立很多socket连接，也会导致奇怪的错误发生。单一Jedis实例不是线程安全的。为了避免这些问题，可以使用JedisPool, JedisPool是一个线程安全的网络连接池。可以用JedisPool创建一些可靠Jedis实例，可以从池中拿到Jedis的实例。这种方式可以解决那些问题并且会实现高效的性能。

初始化JedisPool

```
JedisPool pool = new JedisPool(new JedisPoolConfig(), "localhost");
```

可以以静态的方式处理以上代码，它是线程安全的。

JedisPoolConfig包含了许多有用的redis指定的连接池的默认参数。比如，如果一个连接300秒内没有任何的返回Jedis将关闭这个连接。

可以这样使用

```
Jedis jedis = pool.getResource();
try {
    //随便做一些对于redis的操作
    jedis.set("foo", "bar");
    String foobar = jedis.get("foo");
    jedis.zadd("sose", 0, "car"); jedis.zadd("sose", 0, "bike");
    Set<String> sose = jedis.zrange("sose", 0, -1);
} finally {
    //这里很重要，一旦拿到的jedis实例使用完毕，必须要返还给池中
    pool.returnResource(jedis);
}
//程序关闭时，需要调用关闭方法
pool.destroy();
```

