

在服务器开发的过程中，往往会有一些对象，它的创建和初始化需要的时间比较长，比如数据库连接，网络IO，大数据对象等。在大量使用这些对象时，如果不采用一些技术优化，就会造成一些不可忽略的性能影响。一种办法就是使用对象池，每次创建的对象并不实际销毁，而是缓存在对象池中，下次使用的时候，不用再重新创建，直接从对象池的缓存中取即可。为了避免重新造轮子，我们可以使用优秀的开源对象池化组件apache-common-pool2，它对对象池化操作进行了很好的封装，我们只需要根据自己的业务需求重写或实现部分接口即可，使用它可以快速的创建一个方便，简单，强大对象连接池管理类。

common-pool2简介：

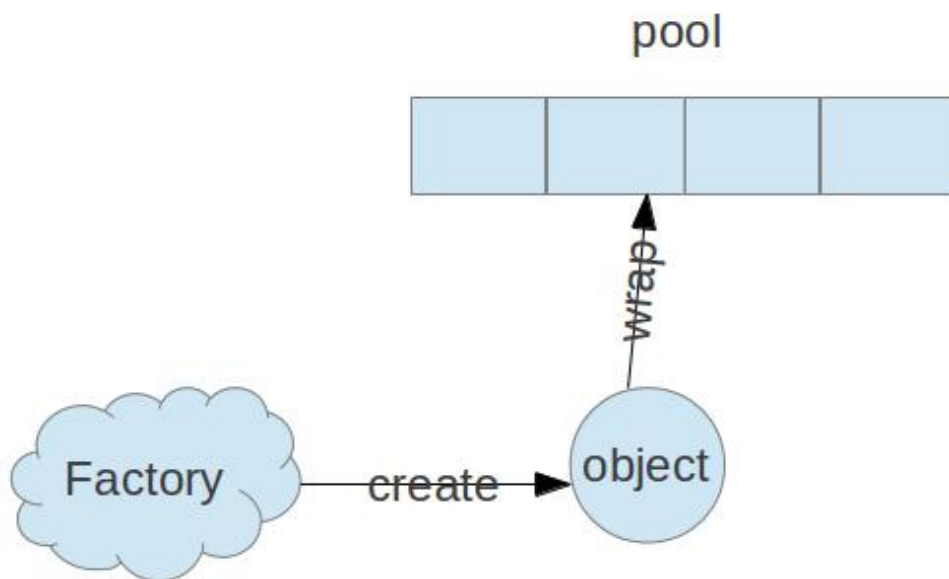
它主要由3个大的部分组成：

ObjectPool接口：专门实现对象存取和状态管理的池实现。我们直接操作的线程池就是定义在这里。值得注意的一点是这里定义的只是怎么来获取以及释放对象等操作，至于具体对象是怎么创建的，一般都通过独立的一个PooledObjectFactory来操心了。

PooledObject：这是commons-pools里比较有意思的一个类族。是对需要放到池里对象的一个包装类。添加了一些附加的信息，比如说状态信息，创建时间，激活时间，关闭时间等。这些添加的信息方便pool来管理和实现一些特定的操作。

PooledObjectFactory：如我们前面所讨论的，管理具体对象的状态，比如创建，初始化，验证对象状态和销毁对象。

他们这三者就构成了一个object pool的基本框架。他们的关系可以用如下的一个图来描述：



ObjectPool类族

common-pool2使用：

GenericObjectPool：继承于**ObjectPool**接口，是对象池实现的核心类，它实现了对对象池的管理，是一个基本的对象池实现，一般情况下，我们可以直接使用。在使用这个类的时候，我们需要传入两个重要的参数：**GenericObjectPoolConfig**类和**PooledObjectFactory**接口的实现

下面给出使用实例：

- TrackerServerPool：

自定义的对象池，通过这个对象池获取对象，归还对象

引用属性 GenericObjectPool对象池