org.apache.commons.pool2

Interface PooledObjectFactory<T>

Type Parameters:
T - Type of element managed in this factory.

All Known Implementing Classes:
BasePooledObjectFactory

public interface PooledObjectFactory<T> 一个"定义对象池中实例的生命周期方法"的接口

An interface defining life-cycle methods for instances to be served by an ObjectPool.

By contract, when an ObjectPool delegates to a PooledObjectFactory, 池对象工厂的职责：

1. makeObject() is called whenever a new instance is needed. 需要新实例
2. activateObject(org.apache.commons.pool2.PooledObject) is invoked on every instance that has been passivated before it is borrowed from the pool.
3. validateObject(org.apache.commons.pool2.PooledObject) may be invoked on activated instances to make sure they can be borrowed from the pool. validateObject(org.apache.commons.pool2.PooledObject) may also be used to test an instance being returned to the pool before it is passivated. It will only be invoked on an activated instance.
4. passivateObject(org.apache.commons.pool2.PooledObject) is invoked on every instance when it is returned to the pool.
5. destroyObject(org.apache.commons.pool2.PooledObject) is invoked on every instance when it is being "dropped" from the pool (whether due to the response from validateObject(org.apache.commons.pool2.PooledObject), or for reasons specific to the pool implementation.) There is no guarantee that the instance being destroyed will be considered active, passive or in a generally consistent state.

必须是线程安全的

PooledObjectFactory must be thread-safe. The only promise an ObjectPool makes is that the same instance of an object will not be passed to more than one method of a PoolableObjectFactory at a time.

相同的对象实例不会在同一时刻被传入到多个方法中

While clients of a KeyedObjectPool borrow and return instances of the underlying value type V, the factory methods act on instances of PooledObject<V>. These are the object wrappers that pools use to track and maintain state information about the objects that they manage.

Since:
2.0

Version:
$Revision: 1333925 $

See Also:
ObjectPool

Method Summary

void	activateObject(PooledObject<T> p) 重新初始化返回给对象池的池对象 Reinitialize an instance to be returned by the pool.
void	destroyObject(PooledObject<T> p) 销毁对象池中不再需要的实例 Destroys an instance no longer needed by the pool.
PooledObject<T>	makeObject() 创建一个能被保存到对象池的实例，并将它包装成一个池对象，以便由对象池管理。 Create an instance that can be served by the pool and wrap it in a PooledObject to be managed by the pool.
void	passivateObject(PooledObject<T> p) 未初始化可以返回给"空闲对象池"的池对象 Uninitialize an instance to be returned to the idle object pool.
boolean	validateObject(PooledObject<T> p) 确保可以返回给"对象池"的池对象实例是安全的 Ensures that the instance is safe to be returned by the pool.

Method Detail

makeObject 制造对象

创建一个能被保存到对象池的实例，并将它包装成一个池对象(PooledObject)，以便由对象池管理。

```
PooledObject<T> makeObject()
                           throws Exception
```

Create an instance that can be served by the pool and wrap it in a PooledObject to be managed by the pool.

Returns:
a PooledObject wrapping an instance that can be served by the pool

Throws:
Exception - if there is a problem creating a new instance, this will be propagated to the code requesting an object.

---

destroyObject 销毁对象

```
void destroyObject(PooledObject<T> p)
                   throws Exception
```

Destroys an instance no longer needed by the pool. 销毁对象池中不再需要的实例

It is important for implementations of this method to be aware that there is no guarantee about what state obj will be in and the implementation should be prepared to handle unexpected errors.

Also, an implementation must take in to consideration that instances lost to the garbage collector may never be destroyed.

必须加以考虑：没有被垃圾收集器回收的实例，可能永远不会被销毁！

Parameters:
p - a PooledObject wrapping the instance to be destroyed

Throws:
Exception - should be avoided as it may be swallowed by the pool implementation.

See Also:
validateObject(org.apache.commons.pool2.PooledObject), ObjectPool.invalidateObject(T)

---

validateObject 验证对象

```
boolean validateObject(PooledObject<T> p)
```

Ensures that the instance is safe to be returned by the pool.

确保可以返回给"对象池"的池对象实例是安全的

Parameters:
p - a PooledObject wrapping the instance to be validated

Returns:
false if obj is not valid and should be dropped from the pool, true otherwise.

如果对象不是有效的，同时应该从对象池中驱除，则返回false；

---

activateObject 激活对象

```
void activateObject(PooledObject<T> p)
                    throws Exception
```

Reinitialize an instance to be returned by the pool. 重新初始化返回给对象池的池对象

Parameters:
p - a PooledObject wrapping the instance to be activated

Throws:
Exception - if there is a problem activating obj, this exception may be swallowed by the pool.

See Also:
destroyObject(org.apache.commons.pool2.PooledObject)

---

passivateObject 消极对象

```
void passivateObject(PooledObject<T> p)
                     throws Exception
```

Uninitialize an instance to be returned to the idle object pool.

确保可以返回给"对象池"的池对象实例是安全的

Parameters:
p - a PooledObject wrapping the instance to be passivated

Throws:

Exception - if there is a problem passivating obj, this exception may be swallowed by the pool.

See Also:

destroyObject(org.apache.commons.pool2.PooledObject)