

org.apache.commons.pool2

Interface ObjectPool<T>

Type Parameters:

T - [Type of element pooled in this pool.](#) 该池中的元素类型

All Known Implementing Classes:

[BaseObjectPool](#), [GenericObjectPool](#), [ProxiedObjectPool](#), [SoftReferenceObjectPool](#)

public interface ObjectPool<T>

A pooling simple interface.

Example of use: 使用示例：

```
Object obj = null;

try {
    obj = pool.borrowObject();
    try {
        //...use the object...
    } catch(Exception e) {
        // invalidate the object
        pool.invalidateObject(obj);
        // do not return the object to the pool twice
        obj = null;
    } finally {
        // make sure the object is returned to the pool
        if(null != obj) {
            pool.returnObject(obj);
        }
    }
} catch(Exception e) {
    // failed to borrow an object
}
```

See [BaseObjectPool](#) for a simple base implementation.

Since: 2.0

Version: \$Revision: 1566605 \$

See Also: [PooledObjectFactory](#), [KeyedObjectPool](#), [BaseObjectPool](#)

Method Summary	
void	addObject() 使用工厂创建一个对象，初始化它，并将它放入空闲对象池中。 Create an object using the factory or other implementation dependent mechanism, passivate it , and then place it in the idle object pool .
T	borrowObject() 从该池中获得一个实例。 Obtains an instance from this pool.
void	clear() 清除池中的任何空闲对象，并释放任何相关的资源。 Clears any objects sitting idle in the pool, releasing any associated resources (optional operation).
void	close() 关闭该池，并释放任何与它相关的资源。 Close this pool, and free any resources associated with it.
int	getNumActive() 返回该池当前被借用的实例数量。 Return the number of instances currently borrowed from this pool.

int	<code>getNumIdle()</code> 返回该池中当前空闲的实例数量。 Return the number of instances <u>currently idle in this pool.</u>
void	<code>invalidateObject(T obj)</code> 标记这个对象为失效状态。 <u>Invalidates an object from the pool.</u>
void	<code>returnObject(T obj)</code> 返回这个实例到池中。 <u>Return an instance to the pool.</u>

Method Detail

borrowObject

[`T borrowObject\(\)`](#)
throws [Exception](#),
[NoSuchElementException](#),
[IllegalStateException](#)

实例可能从makeObject()方法新创建，或是先前的空闲对象，然后使用activateObject(PooledObject)方法激活，然后使用validateObject(PooledObject)方法校验有效性。

Obtains an instance from this pool. 从该池中获得一个实例。

Instances returned from this method will have been either newly created with [PooledObjectFactory.makeObject\(\)](#) or will be a previously idle object and have been activated with [PooledObjectFactory.activateObject\(org.apache.commons.pool2.PooledObject\)](#) and then validated with [PooledObjectFactory.validateObject\(org.apache.commons.pool2.PooledObject\)](#).

By contract, clients must return the borrowed instance using [returnObject\(T\)](#), [invalidateObject\(T\)](#), or a related method as defined in an implementation or sub-interface.

The behaviour of this method when the pool has been exhausted is not strictly specified (although it may be specified by implementations).

Returns: 按约定，客户端必须使用returnObject(T)、invalidateObject(T)
an instance from this pool. 归还借来的实例。

Throws:
[IllegalStateException](#) - after [close](#) has been called on this pool.
[Exception](#) - when [PooledObjectFactory.makeObject\(\)](#) throws an exception.
[NoSuchElementException](#) - when the pool is exhausted and cannot or will not return another instance.

returnObject

[`void returnObject\(T obj\)`](#)
throws [Exception](#)

Return an instance to the pool. By contract, obj must have been obtained using [borrowObject\(\)](#) or a related method as defined in an implementation or sub-interface.

Parameters:
obj - a [borrowed](#) instance to be returned.

Throws:
[IllegalStateException](#) - if an attempt is made to return an object to the pool that is in any state other than allocated (i.e. borrowed). Attempting to return an object more than once or attempting to return an object that was never borrowed from the pool will trigger this exception.
[Exception](#) - if an instance cannot be returned to the pool

invalidateObject

[`void invalidateObject\(T obj\)`](#)
throws [Exception](#)

Invalidates an object from the pool.

By contract, obj **must** have been obtained using [borrowObject\(\)](#) or a related method as defined in an implementation or sub-interface.

This method should be used when an object that has been borrowed is determined (due to an exception or other problem) to be invalid.

Parameters:

obj - a [borrowed](#) instance to be disposed.

Throws:

[Exception](#) - if the instance cannot be invalidated

addObject

void addObject()

throws [Exception](#),
[IllegalStateException](#),
[UnsupportedOperationException](#)

Create an object using the [factory](#) or other implementation dependent mechanism, passivate it, and then place it in the [idle](#) object pool. addObject is useful for "pre-loading" a pool with [idle](#) objects. (Optional operation).

当需要预加载一个池的空闲对象时，这个方法就特别有用！

Throws:

[Exception](#) - when [PooledObjectFactory.makeObject\(\)](#) fails.
[IllegalStateException](#) - after [close\(\)](#) has been called on this pool.
[UnsupportedOperationException](#) - when this pool cannot add new idle objects.

getNumIdle

int getNumIdle()

Return the number of instances [currently idle in this pool](#). This may be considered an approximation of the number of objects that can be [borrowed](#) without creating any new instances. Returns a negative value if this information is not available.

Returns:

the number of instances currently idle in this pool.

getNumActive

int getNumActive()

Return the number of instances [currently borrowed from this pool](#). Returns a negative value if this information is not available.

Returns:

the number of instances currently borrowed from this pool.

clear

void clear()

throws [Exception](#),
[UnsupportedOperationException](#)

Clears any objects sitting [idle](#) in the pool, releasing any associated resources (optional operation). [Idle objects cleared must be PooledObjectFactory.destroyObject\(PooledObject\)](#).

Throws:

[UnsupportedOperationException](#) - if this implementation does not support the operation
[Exception](#) - if the pool cannot be cleared

close

void close()

Close [this pool](#) and free any resources associated with it.

Calling [addObject\(\)](#) or [borrowObject\(\)](#) after invoking this method on a pool will cause them to throw an [IllegalStateException](#).

Implementations should silently fail if not all resources can be freed.
