edu.mit.csail.aeolus.api

Class AeolusQueue<T extends java.io.Serializable>

```
java.lang.Object
  └ edu.mit.csail.aeolus.api.AeolusShared
      └ edu.mit.csail.aeolus.api.AeolusQueue<T>
```

Type Parameters:

T -

All Implemented Interfaces:

AeolusSafe

```
public final class AeolusQueue<T extends java.io.Serializable>extends AeolusShared
```

This behaves like the AeolusBox class but is a queue of objects instead.

Field Summary

Fields inherited from class edu.mit.csail.aeolus.api.AeolusShared
sharedLastEid

Constructor Summary

AeolusQueue() Construct an empty AeolusQueue with the current thread's labels.	
AeolusQueue(AeolusLabel sLabel, AeolusLabel iLabel) Construct an empty AeolusQueue with the specified labels.	

Method Summary

T	**dequeue**() This method returns the object at the head of the queue.
T	**dequeueNoWait**() This call will wait for the lock on the queue before trying to remove a potentially empty head (it will not block if the queue is empty, though).
void	**enqueue**(T obj) Adds a copy of obj to the shared queue.

Methods inherited from class edu.mit.csail.aeolus.api.AeolusShared

`getIntegrityLabel, getSecrecyLabel`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

AeolusQueue

```
public AeolusQueue()
```

Construct an empty AeolusQueue with the current thread's labels.

AeolusQueue

```
public AeolusQueue(AeolusLabel sLabel,
                   AeolusLabel iLabel)
            throws InfoFlowControlException
```

Construct an empty AeolusQueue with the specified labels.

Throws:

`InfoFlowControlException` - if thread's labels are more constraining than those provided.

Method Detail

dequeue

```
public T dequeue()
                                    throws InfoFlowControlException
```

This method returns the object at the head of the queue. (no copying is needed!) It will block until an object is available.

Returns:
dequeued element
Throws:
`InfoFlowControlException` - if thread's labels are not equal to the queue's labels.

---

dequeueNoWait

```
public T dequeueNoWait()
                                        throws InfoFlowControlException
```

This call will wait for the lock on the queue before trying to remove a potentially empty head (it will not block if the queue is empty, though).

Returns:
the object at the head of the shared queue or null if there is no head. (no copying is needed!)
Throws:
`InfoFlowControlException` - if thread's labels are not equal to the queue's labels.

---

enqueue

```
public void enqueue(T obj)
            throws InfoFlowControlException,
                   java.lang.NullPointerException
```

Adds a copy of obj to the shared queue. requires obj != null

Throws:
`java.lang.NullPointerException` - - if obj is null
`InfoFlowControlException` - if queue's labels are more constrained that the thread's.
