edu.mit.csail.aeolus.api

Class AeolusSequence<T extends AeolusSafe & java.io.Serializable>

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

Type Parameters:

T -

All Implemented Interfaces:

AeolusSafe

```
public final class AeolusSequence<T extends AeolusSafe & java.io.Serializable>extends java.lang.Objectimplements AeolusSafe
```

This class is used to create immutable (non-linked) list share-able objects in Aeolus. The elements of this container object are also required to be "safe-to-share" ie immutable as well. So, no copying of elements is needed. "T extends AeolusSafe" causes no copying: i.e., ClassLoader will complain if T is not AeolusSafe.

Constructor Summary

AeolusSequence() Construct an empty AeolusSequence.	
AeolusSequence(java.util.Collection<T> al) Construct an AeolusSequence given a Collection.	

Method Summary

AeolusSequence<T>	**add**(T o) The add method creates an new AeolusSequence which is a copy of this with a new element added onto the end of the sequence.
boolean	**equals**(AeolusSequence sq) The equals method returns true if this sequence's elements are equal to the elements of sq and otherwise returns false.
T	**get**(int index) The get method returns the element at index in the AeolusSequence
boolean	**isEmpty**() The empty method returns true if the AeolusSequence is empty and otherwise returns false
AeolusSequence<T>	**remove**(int index) The remove method creates an new AeolusSequence.
AeolusSequence<T>	**set**(int index, T o) The set method creates an new AeolusSequence.
int	**size**() The size method returns the number of elements in the AeolusSequence

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

AeolusSequence

public **AeolusSequence**()

Construct an empty AeolusSequence.

AeolusSequence

public **AeolusSequence**(java.util.Collection<T> al)

Construct an AeolusSequence given a Collection.

Method Detail

add

```
public AeolusSequence<T> add(T o)
```

The add method creates an new AeolusSequence which is a copy of this with a new element added onto the end of the sequence.

---

equals

```
public boolean equals(AeolusSequence sq)
```

The equals method returns true if this sequence's elements are equal to the elements of sq and otherwise returns false.

---

get

```
public T get(int index)
```

The get method returns the element at index in the AeolusSequence

Throws:

`java.lang.IndexOutOfBoundsException` - if index < 0 or >= size

---

isEmpty

```
public boolean isEmpty()
```

The empty method returns true if the AeolusSequence is empty and otherwise returns false

---

remove

```
public AeolusSequence<T> remove(int index)
```

The remove method creates an new AeolusSequence. The elements of the new AeolusSequence are the same as the elements of the original AeolusSequence except there is one fewer element and the original element at index is missing.

Throws:

`java.lang.IndexOutOfBoundsException` - if index < 0 or >= size

---

set

```
public AeolusSequence<T> set(int index,
                             T o)
```

The set method creates an new AeolusSequence. The elements of the new AeolusSequence are the same as the elements of the original AeolusSequence except that the o replaces the old element at index.

Throws:

`java.lang.IndexOutOfBoundsException` - if index < 0 or >= size

---

size

```
public int size()
```

The size method returns the number of elements in the AeolusSequence

---