

JAVA.UTIL.TREESET.TAILSET METHOD

http://www.tutorialspoint.com/java/util/treeset_tailset.htm

Copyright © tutorialspoint.com

Description

The **tailSet** method is used to return a view of the portion of this set whose elements are greater than or equal to fromElement.

Declaration

Following is the declaration for **java.util.TreeSet.tailSet** method.

```
public SortedSet<E> tailSet(E fromElement)
```

Parameters

- **fromElement** -- This is the low endpoint *inclusive* of the returned set.

Return Value

The method call returns a view of the portion of this set whose elements are greater than or equal to fromElement.

Exception

- **ClassCastException** -- This is thrown if fromElement and toElement cannot be compared to one another using this set's comparator.
- **NullPointerException** -- This is thrown if fromElement or toElement is null and this set uses natural ordering, or its comparator does not permit null elements.
- **IllegalArgumentException** -- This is thrown if fromElement is greater than toElement; or if this set itself has a restricted range, and fromElement or toElement lies outside the bounds of the range.

Example

The following example shows the usage of java.util.TreeSet.tailSet method.

```
package com.tutorialspoint;

import java.util.TreeSet;
import java.util.Iterator;

public class TreeSetDemo {
    public static void main(String[] args) {
        // creating a TreeSet
        TreeSet <Integer>treeadd = new TreeSet<Integer>();
        TreeSet <Integer>treetailset = new TreeSet<Integer>();

        // adding in the tree set
        treeadd.add(1);
        treeadd.add(2);
        treeadd.add(3);
        treeadd.add(4);
        treeadd.add(5);
        treeadd.add(6);
        treeadd.add(7);
        treeadd.add(8);

        // creating tail set
        treetailset=(TreeSet)treeadd.tailSet(4);

        // create iterator
```

```
Iterator iterator;  
iterator = treetailset.iterator();  
  
// displaying the Tree set data  
System.out.println("Tree tail set data: ");  
while (iterator.hasNext()){  
    System.out.println(iterator.next() + " ");  
}  
}
```

Let us compile and run the above program, this will produce the following result.

```
Tree tail set data:  
4  
5  
6  
7  
8
```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js