

JAVA.UTIL.ARRAYS.BINARYSEARCH METHOD

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

Copyright © tutorialspoint.com

Description

The **java.util.Arrays.binarySearch***byte[] a, int fromIndex, int toIndex, byte key* method searches a range of the specified array of bytes for the specified value using the binary search algorithm. The range must be sorted before making this call. If it is not sorted, the results are undefined.

Declaration

Following is the declaration for **java.util.Arrays.binarySearch** method

```
public static int binarySearch(byte[] a, int fromIndex, int toIndex, byte key)
```

Parameters

- **a** -- This is the array to be searched.
- **fromIndex** -- This is the index of the first element *inclusive* to be searched.
- **toIndex** -- This is the index of the last element *exclusive* to be searched.
- **key** -- This is the value to be searched for.

Return Value

This method returns index of the search key, if it is contained in the array, else it returns *-(insertionpoint - 1)*. The insertion point is the point at which the key would be inserted into the array; the index of the first element in the range greater than the key, or toIndex if all elements in the range are less than the specified key.

Exception

- **IllegalArgumentException** -- if fromIndex > toIndex
- **ArrayIndexOutOfBoundsException** -- if fromIndex < 0 or toIndex > a.length

Example

The following example shows the usage of java.util.Arrays.binarySearch method.

```
package com.tutorialspoint;

import java.util.Arrays;

public class ArrayDemo {

    public static void main(String[] args) {

        // initializing unsorted byte array
        byte byteArr[] = {10, 20, 15, 22, 35};

        // sorting array
        Arrays.sort(byteArr);

        // let us print all the elements available in list
        System.out.println("The sorted byte array is:");
        for (byte number : byteArr) {
            System.out.println("Number = " + number);
        }

        // entering the value to be searched
        byte searchVal = 35;
```

```
// entering the range of index
int retVal = Arrays.binarySearch(byteArr,2,5,searchVal);

System.out.println("The index of element 35 is : " + retVal);
}
}
```

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

```
The sorted byte array is:
Number = 10
Number = 15
Number = 20
Number = 22
Number = 35
The index of element 35 is : 4
```

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