

JAVA.UTIL.ARRAYS.FILL METHOD

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

Copyright © tutorialspoint.com

Description

The **java.util.Arrays.fill** *byte[] a, int fromIndex, int toIndex, byte val* method assigns the specified byte value to each element of the specified range of the specified array of bytes. The range to be filled extends from index fromIndex, inclusive, to index toIndex, exclusive.

If fromIndex == toIndex, the range to be filled is empty.

Declaration

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

```
public static void fill(byte[] a, int fromIndex, int toIndex, byte val)
```

Parameters

- **a** -- This is the array to be filled.
- **fromIndex** -- This is the index of the first element *inclusive* to be filled with the specified value.
- **toIndex** -- This is the index of the last element *exclusive* to be filled with the specified value.
- **val** -- This is the value to be stored in all elements of the array.

Return Value

This method does not return any value.

Exception

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

Example

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

```
package com.tutorialspoint;

import java.util.Arrays;

public class ArrayDemo {

    public static void main(String[] args) {

        // initializing byte array
        byte arr[] = new byte[] {1, 6, 3, 2, 9};

        // let us print the values
        System.out.println("Actual values: ");
        for (byte value : arr) {
            System.out.println("Value = " + value);
        }

        // using fill for placing 64 from index 2 to 4
        // converting int to byte
        Arrays.fill(arr, 2, 4, (byte)64);

        // let us print the values
        System.out.println("New values after using fill() method: ");
        for (byte value : arr) {
```

```
System.out.println("Value = " + value);  
}  
}  
}
```

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

```
Actual values:  
Value = 1  
Value = 6  
Value = 3  
Value = 2  
Value = 9  
New values after using fill() method:  
Value = 1  
Value = 6  
Value = 64  
Value = 64  
Value = 9
```

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