

JAVA.MATH.BIGDECIMAL.SETSCALE METHOD

http://www.tutorialspoint.com/java/math/bigdecimal_setscale.htm

Copyright © tutorialspoint.com

Description

The **java.math.BigDecimal.setScale(int newScale)** returns a BigDecimal whose scale is the specified value, and whose value is numerically equal to this BigDecimal's. Throws an ArithmeticException if this is not possible.

This call is typically used to increase the scale, in which case it is guaranteed that there exists a BigDecimal of the specified scale and the correct value. The call can also be used to reduce the scale if the caller knows that the BigDecimal has sufficiently many zeros at the end of its fractional part *i. e. , factorsoftenintstegervalue* to allow for the rescaling without changing its value.

This method returns the same result as the two-argument versions of setScale, but saves the caller the trouble of specifying a rounding mode in cases where it is irrelevant.

Since BigDecimal objects are immutable, calls of this method do not result in the original object being modified, contrary to the usual convention of having methods named setX mutate field X. Instead, setScale returns an object with the proper scale; the returned object may or may not be newly allocated

Declaration

Following is the declaration for **java.math.BigDecimal.setScale** method

```
public BigDecimal setScale(int newScale)
```

Parameters

- **newScale** - scale of the BigDecimal value to be returned

Return Value

This method returns a BigDecimal whose scale is the specified value, and whose unscaled value is determined by multiplying or dividing this BigDecimal's unscaled value by the appropriate power of ten to maintain its overall value

Exception

- **ArithmeticException** - if the specified scaling operation would require rounding

Example

The following example shows the usage of math.BigDecimal.setScale method

```
package com.tutorialspoint;

import java.math.*;

public class BigDecimalDemo {

    public static void main(String[] args) {

        // create 2 BigDecimal Objects
        BigDecimal bg1, bg2;

        bg1 = new BigDecimal("123.126");

        // set scale of bg1 to 6 in bg2
        bg2 = bg1.setScale(6);

        String str = "The value of " +bg1+ " after changing the scale to 6 is
```

```
        " +bg2;  
  
    // print bg2 value  
        System.out.println( str );  
    }  
}
```

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

The value of 123.126 after changing the scale to 6 is 123.126000
Loading [Mathjax]/jax/output/HTML-CSS/jax.js