

JAVA.MATH.BIGDECIMAL.SETSCALE METHOD

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

Copyright © tutorialspoint.com

Description

The **java.math.BigDecimal.setScale(int newScale, int roundingMode)** 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.

If the scale is reduced by the operation, the unscaled value must be divided *rather than multiplied*, and the value may be changed. Here, the specified rounding mode is applied to the division.

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, int roundingMode)
```

Parameters

- **newScale** - scale of the BigDecimal value to be returned
- **roundingMode** - the rounding mode to apply

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 roundingMode == ROUND_UNNECESSARY and the specified scaling operation would require rounding.
- **IllegalArgumentException** - if roundingMode does not represent a valid rounding mode

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("123.12678");

        // set scale of bg1 to 2 in bg2
        // 0 specifies ROUND_UP
        bg2 = bg1.setScale(2, 0);
```

```
String str = bg1 + " after changing the scale to 2 and rounding  
                is " +bg2;  
  
// print bg2 value  
    System.out.println( str );  
}  
}
```

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

123.12678 after changing the scale to 2 and rounding is 123.13

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