

JAVA.LANG.CLASS.GETMETHOD METHOD

http://www.tutorialspoint.com/java/lang/class_getmethod.htm

Copyright © tutorialspoint.com

Description

The **java.lang.Class.getMethod** returns a Method object that reflects the specified public member method of the class or interface represented by this Class object. The **name** parameter is a String specifying the simple name of the desired method.

The **parameterTypes** parameter is an array of Class objects that identify the method's formal parameter types, in declared order. If **parameterTypes** is null, it is treated as if it were an empty array. .

Declaration

Following is the declaration for **java.lang.Class.getMethod** method

```
public Method getMethod(String name, Class<?>... parameterTypes) throws  
NoSuchMethodException, SecurityException
```

Parameters

- **name** -- This is the name of the method.
- **parameterTypes** -- This is the list of parameters.

Return Value

This method returns the Method object that matches the specified name and parameterTypes.

Exception

- **NoSuchMethodException** -- if a matching method is not found or if the name is <init> or <clinit>.
- **NullPointerException** -- If name is null
- **SecurityException** -- If a security manager, s, is present.

Example

The following example shows the usage of java.lang.Class.getMethod method.

```
package com.tutorialspoint;  
  
import java.lang.reflect.*;  
  
public class ClassDemo {  
  
    public static void main(String[] args) {  
  
        ClassDemo cls = new ClassDemo();  
        Class c = cls.getClass();  
  
        try {  
            // parameter type is null  
            Method m = c.getMethod("show", null);  
            System.out.println("method = " + m.toString());  
        }  
  
        catch (NoSuchMethodException e) {  
            System.out.println(e.toString());  
        }  
    }  
}
```

```

try {
    // method Long
    Class[] cArg = new Class[1];
    cArg[0] = Long.class;
    Method lMethod = c.getMethod("showLong", cArg);
    System.out.println("method = " + lMethod.toString());
}
catch(NoSuchMethodException e) {
    System.out.println(e.toString());
}

}

public Integer show() {
    return 1;
}

public void showLong(Long l) {
    this.l = l;
}
long l = 78655;
}

```

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

```

method = public java.lang.Integer ClassDemo.show()
method = public void ClassDemo.showLong(java.lang.Long)

```

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