

Extracting Method in Groovy

This section discusses Extract Method refactoring in Groovy.

This refactoring lets you perform the following actions:

- Extract method for a variable.
- Extract method for a list of variables.
- Extract method for one or several statements.

In this section:

- [Examples](#)
- [To extract a method in Groovy](#)

Examples

Before	After
<pre>iii = 6 int kkk = 5 def vv = 6 def gg = 7 println (kkk + iii + (vv +gg))</pre>	<pre>iii = 6 int kkk = 5 def vv = 6 def gg = 7 println(kkk + iii + testMethod(vv, gg)) private int testMethod(int vv, int gg) { return vv + gg }</pre>
<pre>def a = 5 print a def b = 6 def c = 7 print a + b print a + b + c</pre>	<pre>def a = 5 int b, c (b, c) = testMethod(a) private List testMethod(int a) { print a def b = 6; def c = 7 print a + b return [b, c] } print a+b+c</pre>
<pre>static def foo (int i, int j, int k){ def v println(i + j - k) v = 42 if (i > 42) { println("hello!") } else { return v + j } return 239 }</pre>	<pre>static def foo(int i, int j, int k) { def v println(i + j - k) v = 42 return testMethod(i, v, j) } private static int testMethod(int i, int v, int j) { if (i > 42) { println("hello!") } else { return v + j } return 239 }</pre>

To extract a method in Groovy

1. In the editor, select a block of code to be transformed into a method or a function.

The code fragment to form the method does not necessarily have to be a set of statements. It may also be an expression used somewhere in the code.

2. On the main menu or on the context menu of the selection, choose **Refactor | Extract | Method** or press **Ctrl+Alt+M**.
3. In the **Extract Method** dialog box that opens, specify the name of the new method.
4. To return the value of a data type explicitly, select the **Specify return type explicitly** check box.
5. To return a keyword, select the **Use explicit return statement** check box.
6. In the **Parameters** area, do the following:
 - Specify the variables to be passed as method parameters, by selecting/clearing the corresponding check boxes; if a parameter is disabled, a local variable of the corresponding type, with the initial value ... will be created in the extracted method, so that you will have to enter the initializer with an appropriate value manually.
 - Rename the desired parameters, by double-clicking the corresponding parameter lines and entering new names.
7. Check the result in the **Signature Preview** pane and click **OK** to create the method. The selected code fragment will be replaced with a method call. Additionally, IntelliJ IDEA will propose to replace any similar code fragments found within the current class.

See Also

Concepts:

- [Refactoring Source Code](#)

Reference:

- [Extract Method Dialog for Groovy](#)

Web Resources:

- [Developer Community](#) 