

Extracting Parameter in Groovy

This section discusses the [Extract Parameter](#) refactoring for Groovy. This refactoring lets you perform the following actions:

- Creating a new method parameter from the selected expression within a method. All the usages of the method will be auto-updated.
- Adding a parameter to a closure. If there is a variable, associated with the closure, the calls to this variable are replaced with the corresponding expressions.

In this section:

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

Examples

Before	After
<pre>class Cat { Cat cat = new Cat() def makePestOfItself(){ print ("miaou!!!!!!!!!!") } def makeTroubles(){ if (makePest){ makePestOfItself() } } }</pre>	<pre>class Cat { Cat cat = new Cat() def makePestOfItself(String warning){ print (warning) } def makeTroubles(){ if (makePest){ makePestOfItself("miaou!!!!!!!!!!") } } }</pre>
<pre>class Bar { def foo = { print 'H<caret here>ello, world!' } } new Bar().foo() new Bar().foo.call()</pre>	<pre>class Bar { def foo = { String s -> print s } } new Bar().foo('Hello, world!') new Bar().foo.call('Hello, world!')</pre>

To extract a parameter in Groovy

1. In the editor, place the cursor within the expression to be replaced by a parameter.
2. Do one of the following:
 - Press **Ctrl+Alt+P**.
 - Choose **Refactor | Extract | Parameter** in the main menu.
 - Choose **Refactor | Extract | Parameter** from the context menu.
3. In the [Extract Parameter](#) dialog:
 1. Specify the parameter name in the **Name** field.
 2. Choose parameter type, and specify whether you want to declare the new parameter final, and create the overloading method.
 3. Click **OK**.

See Also

Concepts:

- [Refactoring Source Code](#)

Reference:

- [Extract Parameter Dialog for Groovy](#)

Web Resources:

- [Developer Community](#) 