

Edit Template Variables Dialog

File | Settings | IDE Settings | Live Templates - Edit Variables



The dialog opens when you click the **Edit Variables** button in the [Template Text](#) area on the [Live Templates](#) page.

The **Edit Variables** button is enabled only if the template body contains at least one user-defined variable, that is, a variable different from \$END\$ or \$SELECTION\$.

Use this dialog box to create and edit expressions for variables in the selected live template.

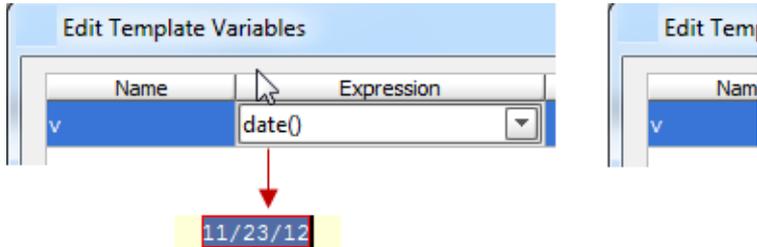
In this topic:

- [Controls](#)
- [Functions](#)

Controls

Item	Description
Name	In this text box, view or edit the variable name in the format <code>\$<variable_name>\$</code> .
Expression	<p>In this text box, specify the expression to have the value of the corresponding template input field calculated automatically.</p> <p>This expression may contain constructs of the following basic types:</p> <ul style="list-style-type: none">■ Predefined functions with possible arguments.■ String constants in double quotes.■ The name of another variable defined in a live template. <p>Type an expression manually or select a predefined function from the drop-down list. The list shows also the number and type of parameters, if any, for the selected function. The available functions are listed alphabetically in the Functions table.</p>
Default value	<p>In this text box, specify the default string to be entered in the corresponding input field of the expanded template, if the expression does not give any result after calculation.</p> <p>Note that a default value of a variable is an expression that can refer to other live template variables. To define the default value as a literal, enclose it in quotation marks.</p>
Skip if defined	Select this check box to have IntelliJ IDEA proceed with the next input field, if the value of the current input field is defined.
Move Up / Move Down	Use these buttons to change the order of variables in the list. The order of variables in the table determines the order in which IntelliJ IDEA will switch between the corresponding input fields when the template is expanded.

Functions

Item	Description
annotated("annotation qname")	Creates a symbol of type with an annotation that resides :
arrayVariable()	Suggests all array variables applicable in the current scope Templates in the <i>iterations</i> group.
capitalize(<name>)	Capitalizes the first letter of the name passed as a paramet
capitalizeAndUnderscore(<String>)	Capitalizes the all letters of a CamelCase name passed as i underscore between the parts. For example, if the string p FooBar, then the function returns FOO_BAR.
castToLeftSideType()	Casts the right-side expression to the left-side expression <i>iterations</i> group to have a single template for generating b Collections.
className()	Returns the name of the current class (the class where th
classNameComplete()	This expression substitutes for the class name completion
clipboard()	Returns the contents of the system clipboard.
complete()	This expression substitutes for the code completion invoca
completeSmart()	This expression substitutes for the smart type completion position.
componentTypeOf (<array variable or array type>)	Returns component type of an array. For example, see the <i>iterations</i> group in the <i>other</i> group.
currentPackage()	Returns the current package name.
date()	Returns the current system date in the specified format. By default, the current date is returned in the default syst specify date format in double quotes, the date will be pres 
decapitalize(<name>)	Replaces the first letter of the name passed as a paramete lowercase letter.
descendantClassEnum(<String>)	Shows the children of the class entered as a string paramet
enum(...)	List of comma-delimited strings suggested for completion a
escapeString()	Escapes the specified string.
expectedType()	Returns the type which is expected as a result of the who template is expanded in the right part of an assignment, ai

Item	Description
fileName()	Returns file name with extension.
fileNameWithoutExtension()	Returns file name without extension.
firstWord(String)	Returns the first word of the string passed as a parameter
groovyScript("groovy code")	Returns Groovy script with the specified code. You can use groovyScript macro with multiple arguments. text that is executed or a path to the file that contains a bound to _1, _2, _3, ..._n variables that are available in Also, _editor variable is available inside the script. This va editor.
guessElementType (<container>)	Makes a guess on the type of elements stored in a java.ut guess, IntelliJ IDEA tries to find the places where the elem extracted from the container.
iterableComponentType(<ArrayOrIterable>)	Returns the type of an iterable component, such as an arr
iterableVariable()	Returns the name of a variable that can be iterated.
lineNumber()	Returns the current line number.
methodName()	Returns the name of the embracing method (where the ter
methodParameters()	Returns the list of parameters of the embracing method (w expanded).
methodReturnType()	Returns the type of the value returned by the current met the template is expanded).
qualifiedClassName()	Returns the fully qualified name of the current class (the c expanded). <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">Clear the Shorten FQ names check box.</div>
rightSideType()	Declares the left-side variable with a type of the right-side <i>iterations</i> group to have a single template for generating b Collections.
snakeCase(String)	Returns snake_case string out of CamelCase string.
subtypes(TYPE)	Returns the subtypes of the type passed as a parameter.
suggestIndexName()	Suggests the name of an index variable. Returns i if there otherwise returns j if there is no such variable in scope, ei
suggestVariableName()	Suggests the name for a variable based on the variable ty according to your code style settings that refer to the var For example, if it is a variable that holds an element within guess on the most reasonable names, also taking into acc being iterated.
suggestFirstVariableName()	Doesn't suggest true, false, this, super.

Item	Description
<code>time()</code>	Returns the current system time.
<code>typeofVariable(VAR)</code>	Returns the type of the variable passed as a parameter.
<code>underscoresToCamelCase</code>	Returns the string passed as a parameter with CamelHump underscores. For example, if the string passed as a parameter is <code>foo_bar</code> , the function returns <code>fooBar</code> .
<code>underscoresToSpaces(VAR)</code>	Returns the string passed as a parameter with spaces substituted for underscores.
<code>user()</code>	Returns the name of the current user.
<code>variableOfType(<type>)</code>	Suggests all variables that may be assigned to the type passed as a parameter. For example <code>variableOfType("java.util.Vector")</code> . If you pass an optional <code>scope</code> parameter, suggests all variables regardless of their types.

Item	Description
<code>JsArrayVariable</code>	Returns JavaScript array name.
<code>jsClassName()</code>	Returns the name of the current JavaScript class.
<code>jsComponentType</code>	Returns JavaScript component type.
<code>jsMethodName()</code>	Returns the name of the current JavaScript method.
<code>jsQualifiedClassName</code>	Returns complete name of a JavaScript class.
<code>jsSuggestIndexName</code>	Returns suggested name for an index.
<code>jsSuggestVariableName</code>	Returns suggested name for a variable.

See Also

Concepts:

- [Live Templates](#)
- [Live Template Variables](#)

Procedures:

- [Creating and Editing Live Templates](#)

Reference:

- [Live Templates](#)

Web Resources:

- [Developer Community](#) 