

Code Style. Groovy

File | Settings | Project Settings | Code Style - Groovy



Use this page to configure [formatting](#) options for Groovy files. View the result in the **Preview** pane on the right.

On this page:

- [Tabs and Indents](#)
- [Spaces](#)
- [Wrapping and Braces](#)
- [Blank Lines](#)
- [Imports](#)
- [Set From](#)

Tabs and indents

Item	Description
Use tab character	<ul style="list-style-type: none">■ If this check box is selected, tab characters are used:<ul style="list-style-type: none">■ On pressing the Tab key■ For indentation■ For code reformatting■ When the check box is cleared, IntelliJ IDEA uses spaces instead of tabs.
Smart tabs	<ul style="list-style-type: none">■ If this check box is selected, IntelliJ IDEA inserts tabs for indentation and reformatting, but fine alignment to a necessary column is done <i>only</i> via spaces without replacing them with tabs. This is done in order to preserve visual representation of the source code, when the Tab Size is changed.■ If this check box is cleared, only tabs are used. This means that a group of spaces that fits the specified tab size is automatically replaced with a tab, which may result in breaking fine alignment. <p>The Smart Tabs check box is available if the Use Tab Character check box is selected.</p>
Tab size	In this text box, specify the number of spaces included in a tab.
Indent	In this text box, specify the number of spaces (or tabs if the Use Tab Character check box is selected) to be inserted for each indent level.
Label indent	In this text box, specify the number of spaces (or tabs if the Use Tab Character check box is selected) to be inserted at the next line before a label statement.
Absolute label indent	If this check box is selected, label indentation is counted as an absolute number of spaces. Otherwise, label indentation is counted relative to previous indent levels.

Spaces

Use this tab to specify where you want spaces in your code. To have IntelliJ IDEA automatically insert a space at a location, select the check box next to this location in the list. The results are displayed in the **Preview** pane.

Wrapping and braces

In this tab, customize the code style options, which IntelliJ IDEA will apply on [reformatting the source code](#). The left-hand pane contains the list of exceptions (**Keep when reformatting**), and placement and alignment options for the various code constructs (lists, statements, operations, annotations, etc.). The right-hand pane shows preview.

Alignment takes precedence over [indentation options](#).

Keep when reformatting

Use the check boxes to configure exceptions that IntelliJ IDEA will make when reformatting the source code. For example, by default, the *Line brakes* check box is selected. If your code contains lines that are shorter than a standard convention, you can convert them by disabling the *Line brakes* check box before you [reformat the source code](#).

Wrapping options

The wrapping style applies to the various code constructs, specified in the left-hand pane (for example, method call arguments, or assignment statements).

Item	Description
Wrapping style	<p>From this drop-down list, select the desired wrapping style:</p> <ul style="list-style-type: none"> ■ <i>Do not wrap</i> - when this option is selected, no special wrapping style is applied. <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p>If this option is selected, the nested alignment and braces settings are ignored.</p> </div> ■ <i>Wrap if long</i> - select this option to have lines going beyond the right margin wrapped with proper indentation. ■ <i>Chop down if long</i> - select this option to have elements in lists that go beyond the right margin wrapped so that there is one element per line with proper indentation. ■ <i>Wrap always</i> - select this option to have all elements in lists wrapped so that there is one element per line with proper indentation.

Alignment options

Item	Description
<character(s)> on next line	Select this check box to have the specified character or characters moved to the next line when the lines are wrapped.
New line after <character>	Select this check box to have the code after the specified character moved to a new line.
Place on new line	Use this check box to have the corresponding statements or characters moved to the next line.

Item	Description
Align when multiline	If this check box is selected, a code construct starts at the same column on each next line. Otherwise, the position of a code construct is determined by the current indentation level.
Special else if treatment	If this check box is selected, <code>else if</code> statements are located in the same line. Otherwise, <code>else if</code> statements are moved to the next line to the corresponding indent level.
Indent case branches	If this check box is selected, the <code>case</code> statement is located at the corresponding indent level. Otherwise, <code>case</code> statement is placed at the same indent level with <code>switch</code> .

Braces placement options

Item	Description
Braces placement style	Use this drop-down list to specify the position of the opening brace in <i>class declarations</i> , <i>method declarations</i> , and other types of declarations. The available options are: <ul style="list-style-type: none"> ■ End of line - select this option to have the opening brace placed at the declaration line end. ■ Next line if wrapped - select this option to have the opening brace placed at the beginning of the line after the multiline declaration line. ■ Next line - select this option to have the opening brace placed at the beginning of the line after the declaration line. ■ Next line shifted - select this option to have the opening brace placed at the line after the declaration line being shifted to the corresponding indent level. ■ Next line each shifted - select this option to have the opening brace placed at the line after the declaration line being shifted to the corresponding indent level, and have the next line shifted to the next indent level as well.
Force braces	From this drop-down list, choose the braces introduction method for <code>if</code> , <code>for</code> , <code>while</code> , and <code>do () while</code> statements. The available options are: <ul style="list-style-type: none"> ■ Do not force - select this option to suppress introducing braces automatically. ■ When multiline - select this option to have braces introduced automatically, if a code construct occupies more than one line. ■ Always - select this check box to have braces always introduced automatically.

Blank lines

Use this tab to define where and how many blank lines you want IntelliJ IDEA to retain and insert in your code after reformatting. For each type of location, specify the number of blank lines to be inserted. The results are displayed in the **Preview** pane.

Item	Description
Keep Maximum Blank Lines	In this area, specify the number of blank lines to be kept after reformatting in the specified locations.
Minimum Blank Lines	<p>In the text boxes in this area, specify the number of blank lines to be present in the specified locations.</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>These settings do not influence the number of blank lines before the first and after the last item.</p> </div>

Imports

Item	Description
General	<p>In this area, configure general import options.</p> <p>Options:</p> <ul style="list-style-type: none"> ■ <i>Use single class import:</i> Select this check box to have IntelliJ IDEA import only a particular class from a package during code generation or import optimization. Otherwise, a statement importing an entire package is inserted. ■ <i>Use fully qualified class names:</i> Select this check box to have IntelliJ IDEA use the fully qualified name of the class to be imported during code generation or import optimization. Otherwise, a normal import statement is inserted. ■ <i>Insert imports for inner classes:</i> Select this check box to have IntelliJ IDEA create imports for the inner classes referenced in your code. ■ <i>Use fully qualified names in Javadoc:</i> Select this check box to have IntelliJ IDEA use a fully qualified class name in Javadoc. Otherwise, a class is imported. ■ <i>Class count to use import with '*':</i> In this text field, specify the number of classes to be imported from a single package until all statements importing a single class are substituted with a statement importing an entire package. ■ <i>Names count to use static import with '*':</i> In this text box, specify the number of classes to be imported from a single package until all statements importing a single class are substituted with a statement importing an entire package.
JSP Imports Layout	<p>In this area, configure how JSP import statements should be organized in your code. The introduced changes are displayed in the Preview pane below.</p> <p>Options:</p> <ul style="list-style-type: none"> ■ <i>Prefer comma separated import list:</i> Select this option to import statements organized in a comma separated list. ■ <i>Prefer one import statement per page directive:</i> Select this option to have one import statement created per line.

Item	Description
Packages to Use Import with '*'	<p>In this area, configure a list of packages to be always imported completely.</p> <p>Options:</p> <ul style="list-style-type: none"> ■ <i>Static</i>: Select this check box, if you want to declare static import for the selected package. ■ <i>Package</i>: In the text fields of this column, specify the packages to be always imported completely. ■ <i>With Subpackages</i>: Select this check box to have all the subpackages of the selected package imported completely. ■ <i>Add Package</i>: Click this button to add a new entry to the list of packages. ■ <i>Remove</i>: Click this button to delete the selected package from the list.
Import Layout	<p>In this area, configure how import statements should be organized in your code. You can set up certain classes to be positioned first, or last, or one after another. Imported classes will be grouped as per their packages and sorted alphabetically within a package.</p> <p>Options:</p> <ul style="list-style-type: none"> ■ <i>Layout static imports separately</i>: If this check box is selected, all static imports will be kept in a separate section. Otherwise, all import statements will be sorted according to the specified layout rules. ■ <i>Static</i>: Select this check box, if you want to declare static import for the selected package. ■ <i>Package</i>: In the text fields of this column, specify the packages to be imported. ■ <i>With Subpackages</i>: Select this check box to have IntelliJ IDEA apply the layout rules to all the subpackages of the selected package. ■ <i>Add Package</i>: Click this button to add a new entry to the list of packages. ■ <i>Add Blank</i>: Click this button to have a blank line inserted after the selected entry, which indicates that a blank line should be inserted between the corresponding import statements. ■ <i>Move Up / Move Down</i>: Click these buttons to move a package or a blank line up or down in the list thus defining the order of import statements. ■ <i>Remove</i>: Click this button to delete the selected package from the list.

Set From

Click this link to reveal the list of languages to be used as the base for the current language code style. So doing, only the settings that are applicable to the current language are taken. All the other settings are not affected.

This link appears in the upper-right corner of the language-specific code style page, when applicable.

Click **Reset** to discard changes and return to the initial set of code style settings.

See Also

Language and Framework-Specific Guidelines:

- [Groovy](#)

Web Resources:

- [Developer Community](#) 