

Format field value

This post function **formats** the current value of a **text field** using a set of **formatting** operations, such as **trimming whitespaces** or performing a **search and replace**.



Configuration

Field

Select the field to be formatted. Available field types are text, versions, labels, select list, radio button, multi-select list and cascading select list.

White space mode

Select whether to edit blanks (or whitespaces). The following options are available (multiple choice is possible):

- Remove all leading white spaces
- Remove all trailing white spaces
- Leave only one white space between words
- Remove all white spaces in the text

Case mode

Select whether to edit cases. The following options are available:

- Do nothing
- Convert all the characters to lower case
- Convert all the characters to upper case
- Capitalize all the white space separated words

Find and replace

Define whether you want to find and replace parts of the text.

You first have to define the **fragments** that you want to **replace** and afterwards specify the **replacement**. Both parameters can be specified using an expression in [Basic text mode](#).

Find mode

Option	Description
Text (case sensitive)	The text to be replaced must exactly match, respecting the case.
Text (ignoring case)	The text to be replaced must match but the case will be ignored.
Regular expression (case sensitive)	Specify the text to be replaced by a regular expression, respecting the case.
Regular expression (ignoring case)	Specify the text to be replaced by a regular expression, ignoring the case.

Replace mode

Option	Description
Replace first occurrence	Only the first occurrence of the text will be replaced.
Replace all occurrences	All occurrences of the text will be replaced.

Additional options

Choose to **update issues immediately** if you want Jira to fire a separate "Issue updated" event for any field change. This will also result in a dedicated **issue history** entry.

Conditional execution

You can **optionally** specify a [logical expression](#) to define the circumstances (or conditions) under which the post function should be executed.

The result of the logical expression must return a boolean value of either:

- **true** the post function will be executed
- **false** the post function will **not** be executed

Using the **conditional operator**, even complex or multi-layered conditions can be constructed.

Make sure to learn more about defining logical expressions and browse through the various **examples** here: [Logical mode](#)



Use cases and examples

Use case	JWT feature	Workflow function	Parser functions	Label
Format the issue's summary according to specified rules		Format field value		STAFF PICK

If you still have questions, feel free to refer to our [support](#) team.