

# Format field value

This function has been **renamed** with the [JWT 3.0](#) release.

Find the new documentation at:

[Format field value](#)

## On this page

- [Purpose](#)
- [Example: Formatting issue summary and replacing contained terms](#)
- [Configuration Parameters](#)
- [Usage Examples](#)
- [Related Features](#)

## Purpose

This post-functions allows to give format to current value of a field using a set of formatting operations

## Example: Formatting issue summary and replacing contained terms

Now, let's see a configuration example for formatting issue **Summary** for removing undesired white spaces, capitalizing words and replacing the word "**Client**" by "**Customer**":

<b>Field to be formatted:</b>	Summary - [Text] ▼	<a href="#">?</a>
<b>Blanks trimming operations:</b>	<input checked="" type="checkbox"/> Remove all leading white spaces <input checked="" type="checkbox"/> Remove all trailing white spaces <input checked="" type="checkbox"/> Leave only one white space between words <input type="checkbox"/> Remove all white spaces in the text	
<b>Case changing operations:</b>	<input type="radio"/> Do nothing <input type="radio"/> Convert all the characters to lower case <input type="radio"/> Convert all the characters to upper case <input checked="" type="radio"/> Capitalize all the whitespace separated words	
<b>Find &amp; Replace operations:</b>	<p><u>Substring to be found:</u></p> <input type="radio"/> literal text <input checked="" type="radio"/> text literal (ignoring case) <input type="radio"/> regular expression <input type="radio"/> regular expression (ignoring case)	
	<p>Client</p> <p>Field code injector: Summary - [Text] - %{00000} ▼</p> <p>Dynamic patterns can be created through insertion of field codes that will be replaced with the corresponding field values.</p>	
	<p><u>Replacement:</u></p> <input type="radio"/> replace first <input checked="" type="radio"/> replace all	
	<p>Customer</p> <p>Field code injector: Summary - [Text] - %{00000} ▼</p> <p>Dynamic patterns can be created through insertion of field codes that will be replaced with the corresponding field values.</p>	
<b>Conditional execution:</b> Optional boolean expression that should be satisfied in order to actually execute the post-function. <a href="#">(Syntax Specification)</a>	<div>1</div> <p>Leave the field empty for executing the post-function unconditionally.   <a href="#">Collection of Examples</a>   [ Line 1 / Col 1 ]</p> <p><u>Logical connectives:</u> and, or and not. Alternatively you can also use &amp;,   and !.</p> <p><u>Comparison operators:</u> =, !=, &gt;, &gt;=, &lt; and &lt;=. Operators in, not in, any in, none in, ~ and != can be used with <i>strings</i>, <i>multi-valued fields</i> and <i>lists</i>.</p> <p><u>Logical literals:</u> true and false. Literal null is used with = and != to check whether a field is initialized, e.g. {00012} != null checks whether <i>Due Date</i> is initialized.</p> <div> <div> <u>String Field Code Injector:</u>            Summary - [Text] - %{00000} ▼         </div> <div> <u>Numeric/Date Field Code Injector:</u>            Original estimate (minutes) - [Number] - {00068} ▼         </div> </div>	

Once configured, post-function looks like this:

Triggers 0	Conditions 1	Validators 1	Post Functions 9
<p><b>The following will be processed after the transition occurs</b>   <a href="#">Add post function</a></p> <ol style="list-style-type: none"> <li>Field <b>Summary</b> will be formatted using the following operations:             <ul style="list-style-type: none"> <li>Leading blanks will be removed.</li> <li>Trailing blanks will be removed.</li> <li>Repeated blanks between words will be removed.</li> <li>All the whitespace separated <b>words</b> will be capitalized.</li> <li>Find all the occurrences of <b>literal text "Client"</b> ignoring the case, and replace with <b>"Customer"</b>.</li> </ul> </li> </ol>			

## Configuration Parameters

## Blanks trimming operations

- 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 changing operations

- Convert all the characters **to lower case**
- Convert all the characters **to upper case**
- **Capitalize** all the whitespace separated words

## Find & Replace (literal strings or regular expressions)

- **Replace first** found substring found
- **Replace all** found substrings

## Supported Field Types

The following field and custom field types can be formatted:

- **Text** fields: Summary, Description, Environment, etc. Also Single-line and Multi-line text custom fields.
  - **Versions** fields: Fix Version/s, Affects Version/s and Version Pickers custom fields (single and multi select).
  - **Labels** fields: issue's Labels field and Labels custom fields.
  - **Selectable fields**: Select List, Multi-Select List, Radio Button
  - **Cascading Select List**
- 

## Usage Examples

## Related Features

- [Copy parsed text to a field](#): with this post-function much more complex text forming operation can be implemented. To do it use **advanced parsing mode**.
- [Parse field for extracting data](#)
- [Regular expression renderer](#)