

findPattern()

This function returns a **text list** with all **substrings** matching a given regular expression.

Syntax

findPattern(text, regex) #Output: Text list

Examples

Parser expression	Description
<pre>%{findPattern("Between 1900 and 2000 world population increase from 1.5 to 6.1 billions.", "\d+(\.\d+)?")}</pre>	This example returns: 1900,2000,1.5,6.1
<pre>%{findPattern(%{issue.somefield}, "(?<=mailto:)([a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+\.[a-zA-Z0-9_-]+)")}</pre>	Extract all email addresses from a text field Assuming %{issue.somefield} is a multi-line text fiel containing email addresses (added as hyperlinks), e.g. <pre> Hello, please use admin@example.com instead of administrator@example.com for future reference. Kind regards, jwt@example.com </pre> this example returns: admin@example.com,administrator@example.com,jwt@example.com
<pre>%{findPattern(%{issue.lastComment}, "(?<=~accountid:).*?(?=\\)")}</pre>	Extract all user mentions from the last comment Assuming %{issue.lastComment} returns, e.g. <pre> Hi [Alina Grant]-accountid:5c90f542efd65d2d3d465d7c! </pre> this example returns: 5c90f542efd65d2d3d465d7c
<pre>%{findPattern(%{issue.lastComment}, "([A-Z][A-Z0-9]+)-\\d+")}</pre>	Extract all issue keys from the last comment Assuming %{issue.lastComment} returns, e.g. <pre> Hi [admin], this issues seems to be related to DEMO-1 and DEMO-2! </pre> this example returns: DEMO-1,DEMO-2

```
%{findPattern(%{issue.
somefield}, "(?
<=\\http://|\\https://).*(?
=\\)")}}
```

Extract all URLs from a text field

Assuming %{issue.somefield} returns, e.g.

The best documentation can be found at [apps.decadis.net|https://apps.decadis.net]. Make sure to check the space for [Jira Workflow Toolbox|https://apps.decadis.net/display/JWT]! If you can't find what you're looking for there, all you have left is [Google|https://www.google.com].

this example returns:

apps.decadis.net,apps.decadis.net/display/JWT,www.google.com

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
text	TEXT	Any given text.
regex	TEXT	A valid regular expression that grabs all substrings to be returned.

Output

This function returns a

If you want to ignore the case, have a look at the [findPatternIgnoreCase\(\)](#) function.

Parser expression	Output
<code>findPatternIgnoreCase("Grass is Green and Sky is Blue.", "red green blue")</code>	<code>["Green", "Blue"]</code>
<code>findPattern("Grass is Green and Sky is Blue.", "red green blue")</code>	<code>[]</code>



Use cases and examples

Use case

No content found.