

findPattern()

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

Syntax

findPattern(text, regularExpression) #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 the following text list: ["1900", "2000", "1.5", "6.1"]
<pre>findPattern(%{issue.somefield}, "(?<=CN=).*?(?=?,OU=Users)")</pre>	Assuming %{issue.somefield} is a multi-line text field containing a list of Active Directory user entries, e.g. <pre>CN=User1,OU=Users,DC=example,DC=com CN=User2,OU=Users,DC=example,DC=com CN=User3,OU=Users,DC=example,DC=com CN=User4,OU=Users,DC=example,DC=com</pre> this example returns the following text list: ["User1", "User2", "User3", "User4"]
Extract all email addresses from a text field <pre>findPattern(%{issue.somefield}, "([a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+\.\[a-zA-Z0-9._-]+)")</pre>	Assuming %{issue.somefield} is a multi-line text field containing email addresses, 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 the following text list: ["admin@example.com", "administrator@example.com", "jwt@example.com"]
Processing an Elements Connect (formerly nFeed)* custom field <pre>findPattern(%{issue.somefield}, "(?<=:\\[\\\" \\\",\\\").*?(?=\\\" \\\",\\\")")</pre> * Applicable for Elements Connect version 6 and above.	Assuming %{issue.somefield} is an Elements Connect (formerly nFeed) custom field containing the keys of the selected options, e.g. <pre>{ "keys": ["KEY-1", "KEY-2", "KEY-3"] }</pre> this example returns the following text list: ["KEY-1", "KEY-2", "KEY-3"]

Extract all user mentions from the last comment <pre>distinct(findPattern(%{issue.lastComment}, "(?<=\\[~].*?(?=\\])"))</pre>	Assuming <code>%{issue.lastComment}</code> returns, e.g. Hi [~demo] and [~user]! this example returns the following text list : ["demo", "user"]
Extract all issue keys from the last comment <pre>findPattern(%{issue.lastComment}, "([A-Z][A-Z0-9]+)-\\d+")</pre>	Assuming <code>%{issue.lastComment}</code> returns, e.g. Hi [~admin], this issues seems to be related to DEMO-1 and DEMO-2! this example returns the following text list : ["DEMO-1", "DEMO-2"]
Extract all URLs from a text field <pre>findPattern(%{issue.somefield}, "(?<=\\[http:\\\\ https:\\\\ \\].*?(?=\\])")</pre>	Assuming <code>%{issue.somefield}</code> 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 the following text list : ["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.
regularExpression	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>	["Green", "Blue"]
<code>findPattern("Grass is Green and Sky is Blue.", "red green blue")</code>	[]



Use cases and examples

Use case	JWT feature	Workflow function	Field type	Automated action	Parser functions
Display reporters of linked Cloud issues			Text		<code>executeRemoteAction()</code> <code>findPattern()</code> <code>toStringList()</code> <code>toString()</code> <code>distinct()</code>
Obtain a value from an Elements Connect custom field		Send email			<code>findPattern()</code> <code>toString()</code>
Evaluate Assets objects	 	Logical validator Logical condition			<code>findPattern()</code> <code>findReplaceAll()</code> <code>replaceAll()</code> <code>toStringList()</code>
Copy excerpted value from new comments				Update field action	<code>findPattern()</code> <code>first()</code>
Send email with the URL of the attachments included in the description		Send email			<code>replaceAll()</code> <code>toString()</code> <code>attachmentUrls()</code> <code>findPattern()</code>
Link issue to issue keys in its description		Create issue link			<code>findPattern()</code> <code>toString()</code>
Create a sub-task for every sub-task closed		Create issue			<code>findReplaceFirst()</code> <code>first()</code> <code>findPattern()</code> <code>toString()</code> <code>sum()</code> <code>toNumber()</code>
Track issues mentioned in comments				Create issue link action	<code>findPattern()</code> <code>issueKeysToIssueList()</code> <code>toString()</code>
Triage issues created by email		Move issue			<code>toUpperCase()</code> <code>toString()</code> <code>findPattern()</code> <code>isInGroup()</code>