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
`findPattern("Between 1900 and 2000 world population increase from 1.5 to 6.1 billions.", "\\d+(\\.\\d+)?")`	This example returns the following **text list**: **["1900", "2000", "1.5", "6.1"]**
`findPattern(%{issue.somefield},"(?<=CN=).*?(?=,OU=Users)")`	Assuming `%{issue.somefield}` is a multi-line text field containing a list of Active Directory user entries, e.g. *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* this example returns the following **text list**: **["User1", "User2", "User3", "User4"]**
Extract all email addresses from a text field `findPattern(%{issue.somefield},"([a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+\\.[a-zA-Z0-9_-]+)")`	Assuming `%{issue.somefield}` is a multi-line text field containing email addresses, e.g. *Hello,* *please use admin@example.com instead of administrator@example.com for future reference.* *Kind regards,* *jwt@example.com* this example returns the following **text list**: **["admin@example.com", "administrator@example.com", "jwt@example.com"]**
Processing an Elements Connect (formerly nFeed)* custom field `findPattern(%{issue.somefield},"(?<=:\\[\"|\",\").*?(?=\")")` * 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. `{ "keys":[ "KEY-1", "KEY-2", "KEY-3" ] }` this example returns the following **text list**: **["KEY-1", "KEY-2", "KEY-3"]**

Extract all user mentions from the last comment `distinct(findPattern(%{issue.lastComment},"(?<=\\[~).*?(?=\\])"))`	Assuming `%{issue.lastComment}` returns, e.g. > *Hi [~demo] and [~user]!* this example returns the following **text list**: **["demo", "user"]**
Extract all issue keys from the last comment `findPattern(%{issue.lastComment},"([A-Z][A-Z0-9]+)-\\d+")`	Assuming `%{issue.lastComment}` 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 `findPattern(%{issue.somefield},"(?<=\\|http:\\/\\/|\\|https:\\/\\/).*?(?=\\])")`	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 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 TEXT LIST

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

Parser expression	Output
`findPatternIgnoreCase("Grass is Green and Sky is Blue.", "red|green|blue")`	**["Green", "Blue"]**
`findPattern("Grass is Green and Sky is Blue.", "red|green|blue")`	**[ ]**

Use cases and examples

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