

jiraExpression()

This function returns the result of a given [Jira expression](#) as a text. In case a number, a list or a logical value is returned by the [Jira expression](#), the returned text of the function `jiraExpression()` can be converted accordingly, e.g. with `toStringList()`.

It unfolds its power when combined with JWT functionality which is not available in Jira expressions and vice versa.

Syntax


```
jiraExpression(expression) #Output: Text
```

Examples

Parser expression	Description
<pre>%{sort(distinct(toStringList(jiraExpression("issue.subtasks.map(s=>s.components.map(c=>c.name))")),ASC)}</pre>	<p>This example returns an ordered list of the unique component names of all sub-tasks of an issue, e.g.</p> <p>Backend, Frontend, UI</p> <p>To convert to text result, the following function is used:</p> <ul style="list-style-type: none">• toStringList()
<pre>%{Hi [~accountid:%{issue.project.lead}], the user %{system.currentUser.displayName} has written %{jiraExpression("issue.comments.filter(c=>c.author.accountId==user. accountId).length")} comments to this issue and created % {jiraExpression("issue.subtasks.filter(s=>s.reporter.accountId==user. accountId).length")} sub-tasks.}</pre>	<p>This example returns:</p> <p>Hi [~accountid:557058:145e0473-5707-439c-80e4-1160dd57f114], the user John Doe has written 3 comments to this issue and created 2 sub-tasks.</p>
<pre>issue.project.lead = system.currentUser ? (toLogicalValue(jiraExpression("issue.votes == 0")) ? "This issue does not seem to be important for anyone" : "There are voters for this issue!") : "Not interesting for you!"</pre>	<p>The function can also easily be used in a logical expression by converting the text result to a logical value (by using the function <code>toLogicalValue()</code>):</p> <p>If this expression runs for a user which is the project lead, this example returns :</p> <p>There are voters for this issue! in case there voters for this issue</p> <p>This topic does not seem to be important for anyone if there are no voters for this issue.</p> <p>In case the current user is not the project lead, only the message Not interesting for you! is returned.</p> <p>To achieve this the following functions is used:</p> <ul style="list-style-type: none">• toLogicalValue()

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
expression	<input type="text" value="TEXT"/>	Any given text which can be evaluated as Jira expression .  The expression is not checked for syntactical correctness when defining it in the parser expression editor .

Output

This function returns a

Variant where you can additionally define a list of issues which can be used within the [Jira expression](#) given as the first parameter. You can access these issues [through an injected context called **issues**](#).

Syntax


```
jiraExpression(expression, issueList) #Output: Text
```

Examples

Parser expression	Description
<pre>%{sort(distinct(toStringList(jiraExpression("issues.flatMap(i=>i. subtasks).flatMap(s=>s.components).map(c=>c.name)", linkedIssues()), ASC)} </pre>	<p>This example returns an ordered list of the unique component names of all sub-tasks of the linked issues, e.g.</p> <p>Backend, Frontend, UI</p> <p>To achieve this the following function is used:</p> <ul style="list-style-type: none">linkedIssues() <p>You have to use flatMap because sub-tasks of a list of issues are returned as a multi list in a Jira expression.</p> <p>Since the function returns a text, it is converted to a text list using toStringList().</p>
<pre>toLogicalValue(jiraExpression("issues.every(c=>c. name=='UI')",subtasks())) </pre>	<p>This example returns <code>true</code> in a logical expression if all subtasks have a component UI</p> <p>To achieve this the following functions are used:</p> <ul style="list-style-type: none">toLogicalValue()subtasks()
<pre>toStringList(issue.components.leads) ~ system. currentUser AND toLogicalValue(jiraExpression("issues.map(i=>i. issueType.name).includes('Task')",linkedIssues())) </pre>	<p>This example returns <code>true</code> in a logical expression if the current user is lead of at least one componnetall subtasks have a component UI</p> <p>To achieve this the following functions are used:</p> <ul style="list-style-type: none">toLogicalValue()linkedIssues()

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
expression	<input type="text" value="TEXT"/>	Any given text which can be evaluated as Jira expression .  The expression is not checked for syntactical correctness when defining it in the parser expression editor .
issueList	<input type="text" value="ISSUE LIST"/>	Any given issue list. Usually this value is retrieved from a function (e.g. linkedIssues() or subtasks()).

Output

This function returns a

Variant where you can additionally define a string of Issue keys which can be used within the [Jira expression](#) given as the first parameter. You can access these issues through an injected context called **issues**.

Syntax


```
jiraExpression(expression, issueKeys) #Output: Text
```

Examples

Parser expression	Description
<pre>%{sort(distinct(toStringList(jiraExpression("issues.map(i=>i.subtasks). flatten().map(s=>s.components.map(c=>c.name)). flatten()", "PU-2, PU-7, PU-120")), ASC)}</pre>	<p>This example returns an ordered list of the unique component names of all sub-tasks of the issues listed in the second parameter, e.g.</p> <p>Backend, Frontend, UI</p> <p>You have to use flatten because sub-tasks of a list of issues are returned as a multi list in a Jira expression.</p> <p>Since the function returns a text, it is converted to a text list using toList().</p>

Additional information

Parameters used in this function




Parameter	Input (data type)	Description
expression	<input type="text" value="TEXT"/>	Any given text which can be evaluated as Jira expression .  The expression is not checked for syntactical correctness when defining it in the parser expression editor .

issueKeys	<input type="text" value="TEXT"/>	A text with a comma-separated list of issue keys .
-----------	-----------------------------------	---

Output

This function returns a

Use cases and examples

Use case	JWT feature	Workflow function	Field type	Automated action	Parser functions
Add request participants		Execute remote action			jiraExpression()
Create a sub-task for each user selected in a User Picker field		Create issue			jiraExpression()
Inform the project manager about an added attachment		Send email			jiraExpression()