

Migrate 'Logical condition'

The [Logical condition of JWT DC](#) is not available in JWT Cloud , but can be implemented easily using a [Jira expression condition](#) .

Since JWT DC field codes are not available in conditions or validators in JWT Cloud, they have to be replaced by their Jira expression equivalent.

Migration steps

1

Add a [Jira expression condition](#).

2

Build a matching [Jira expression](#) depending on the individual configuration. Since the basic concept of JWT parser expressions is strictly different from Jira expressions, it is not possible to migrate them directly.


The following list provides the necessary basics.

Migration details

Fields

The field codes supported by Jira expressions differ from the JWT ones (please see our comprehensive [Field codes](#) section). The can be inserted using the field code injector within the expression editor. For further details about Jira expression field codes please refer to [Atlassian's documentation](#).

Operators

 JWT DC	Jira expression	Notes
=	==	
!=	!=	
<	<	
<=	<=	
>	>	
>=	>=	
~	.includes()	Example: <div>issue.summary.includes("issue")</div>
!~	!...includes()	Example: <div>issue.summary.includes("issue")</div>

IN	.every....includes()	Example: <pre>issue?.subtasks?.every(s=>issue?.links?.map(link => link.linkedIssue.key).includes(s.key))</pre>
NOT IN	! every....includes()	Example: <pre>!issue?.subtasks?.every(s=>issue?.links?.map(link => link.linkedIssue.key).includes(s.key))</pre>
ANY IN	.some()....includes()	Example: <pre>issue?.subtasks?.some(s=>issue?.links?.map(link => link.linkedIssue.key).includes(s.key))</pre>
NONE IN	!....some()....includes()	Example: <pre>!issue?.subtasks?.some(s=>issue?.links?.map(link => link.linkedIssue.key).includes(s.key))</pre>

Functions

Jira expression functions are based on the respective data type and are related to JavaScript notion. Details of the available functions can be found in [Atlassian's documentation](#).

Examples

 JWT DC use case	JWT DC parser expression	Jira expression
Make "Time spent" field required	<pre>{issue.timeSpent} > 0</pre>	<pre>issue?.timeSpent > 0</pre>
Block an Epic's transition until all the issues under that Epic are resolved	<pre>%{issue.issueType} = "Epic" IMPLIES count(filterByResolution (issuesUnderEpic(), "")) = 0</pre>	<pre>issue?.issueType?.name == "Epic" ? issue?.stories?.every(s=>s.resolution != null) : true</pre> <pre>issue?.isEpic ? issue?.stories?.every (s=>s.resolution!= null) : true</pre>
Block a transition until all sub-tasks have certain fields populated	<pre>count(filterByPredicate(subtasks(), {function.issue.dueDate} != null)) = count(subtasks())</pre>	<pre>issue?.subtasks?.every(i=>i.dueDate != null)</pre>

Due to the different architecture, it may happen that the condition gets too complex. This is the case when many fields are checked. The condition cannot be saved, and a corresponding error message will be displayed. If that's the case, the condition has to be split up into two or more.

If you still have questions, feel free to refer to our [support](#) team.