

Examples of String expressions

Text Composition and Format

Expression	Example of returned value
"Current issue was reported on " + %{...created} + " by " + %{...reporterUserFullName} + "."	Current issue was reported on 2014-09-03 19:28 by John Nash.
"Today is " + dayOfTheWeekToString({...currentDateTime}, USER_LOCAL, USER_LANG) + "."	Today is Monday.
"Number of hours since issue creation: " + round(({...currentDateTime} - {...created}) / {HOUR}) + " hours."	Number of hours since issue creation: 75 hours.
"Number of days to due date: " + floor(({...duedate} - {...currentDateTime}) / {DAY}) + " days."	Number of days to due date: 2 days.

On this page

- [Text Composition and Format](#)
- [Issue Selection](#)
- [Working with Fields in Linked Issues and Sub-tasks](#)
- [Logical Constructions](#)

Issue Selection

Expression	Example of returned value
<code>filterByFieldValue(subtasks(), %{...components}, ~, "Component A")</code> or alternatively <code>filterByPredicate(subtasks(), %{...components} ~ "Component A")</code>	Returns an issue list with sub-tasks having " Component A " among its components.
<code>except(subtasks(%{...parentIssueKey}), issueKeysToIssueList(%{...issuekey}))</code> or alternatively <code>filterByPredicate(subtasks(%{...parentIssueKey}), ^%{...issuekey} != %{...issuekey})</code>	Returns an issue list with sibling sub-tasks, i.e., parent's sub-tasks except current issue.
<code>filterByFieldValue(filterByIssueType(getIssuesFromProjects(%{...project}), %{...issuetype}), %{...summary}, =, %{...summary})</code> or alternatively <code>filterByPredicate(getIssuesFromProjects(%{...project}), ^%{...issuetype} = %{...issuetype} AND ^%{...summary} = %{...summary})</code>	Returns an issue list with all issues in the same project as current issue, with same issue type and same summary. Might be used in combination with function count() for creating a validation to avoid issue creation when an issue with same summary already exists in the project and issue type.

Working with Fields in Linked Issues and Sub-tasks

Expression	Example of returned value
------------	---------------------------

<code>filterByCardinality(fieldValue(%{...components}, subtasks()), =, count(subtasks()))</code>	["Component A", "Component B", "Component C"] Returns a string list with the Components present in all sub-tasks of current issue, i. e., those components common to all sub-tasks.
<code>{...duedate} > max(fieldValue({...duedate}, union(linkedIssues("is blocked by"), subtasks())))</code>	Validation to check that: Due Date is greater than latest Due Date among blocking issues and sub-tasks.
<code>count(filterByFieldValue(subtasks(), %{...environment}, =, "") UNION filterByFieldValue(subtasks(), %{...duedate}, =, "")) = 0</code> or alternatively <code>count(filterByPredicate(subtasks(), ^%{...environment} = null OR ^%{...duedate} = null)) = 0</code>	Expression for checking whether all sub-tasks of current issue have fields Due date and Environment set.
<code>count(filterByPredicate(linkedIssues("is Epic of"), ^%{...resolution} != null OR ^{...duedate} = null)) = 0</code>	This validation allows certain transition in Epic's workflow to be executed, only if all the tasks are unresolved and have Due Date set.

On this page

- [Text Composition and Format](#)
- [Issue Selection](#)
- [Working with Fields in Linked Issues and Sub-tasks](#)
- [Logical Constructions](#)

Logical Constructions

Expression	Example of returned value
<code>!(%{...priority} = "Blocker" OR %{...priority} = "Critical") OR {...duedate} != null</code>	Validation for checking that: If Priority is " Blocker " or " Critical " then Due Date must be initialized. It is based on equivalent logical constructions: A implies B = !A OR B
<code>%{...priority} = "Blocker" OR %{...priority} = "Critical" IMPLIES {...duedate} != null</code>	Validation for checking that: If Priority is " Blocker " or " Critical " then Due Date must be initialized. Same as former example but using logical connective IMPLIES .
<code>{...duedate} = null OR {...duedate} >= ({...created} + 2 * {DAY})</code>	Validation for checking: If Due Date is set then it must be equal or grater than Day and Time of Creation plus 2 days.