

Constructing JQL Queries

JIRA 1 Example Clause `project = TEST`

JIRA 2 Example Clause `project = TEST AND assignee in (currentuser())`

Breakdown

Field `project`

Operator `=`

Value `TEST`

Values & Functions `TEST AND assignee = currentuser()`

A simple query in JQL (also known as a 'clause') consists of a field, followed by an operator, followed by one or more values or functions. For example:

Scoping and sorting

Scoping Focusing your query so it pulls the right amount of data so the user sees only the information relevant to the current item at hand.

Sorting Ordering your data such that the most critical set of data is listed first.

Part of a good query is knowing how to get data out of JIRA. The other part is knowing how to engage your team and your customers with the result. Let's start with two concepts that are critical to effective data presentation.

Fields

Assignee Epic Link Resolved

Affected version Filter Sprint

Attachments Fix version Status

Comment Issue key Summary

Component Labels Text

Created Last viewed Time spent

Creator Priority Voter

Description Project Watcher

Due Reporter *custom field*

A field in JQL is a word that represents a Jira field (or a custom field that has already been defined in Jira).

Operators

`=` `!=`

`>` `<`

`>=` `<=`

`~` `!~`

`in` `not in`

`is` `is not`

`was` `was not`

`was in` `was not in`

`changed`

An operator in JQL is one or more symbols or words that compare the value of a field on its left with one or more values (or functions) on its right, such that only true results are retrieved by the clause. Some operators may use the NOT keyword.



Functions

Time	People	Issue
startOfDay/Week/Month/Year	currentLogin()	issueHistory()
endOfDay/Week/Month/Year	currentUser()	openSprints()
lastLogin()		watchedIssues()
now()		myApproval()
		myPending()

A function in JQL appears as a word followed by parentheses, which may contain one or more explicit values or Jira fields.

A function performs a calculation on either specific Jira data or the function's content in parentheses, such that only true results are retrieved by the function, and then again by the clause in which the function is used.

Reserved characters & words

space (" ")	/	a, and, are, as, at,
+	%	be, but, by, for, if,
.	^	in, into, is, it, no,
,	\$	not, of, on, or, s,
;	#	such, t, that, the, their,
?	@	then, there, these, they,
	[this, to, was, will, with
*]	

When using these common characters or words in queries, you need to:

1. Surround them with quote-marks. You can use either single quote-marks (') or double quote-marks (") eg. text ~ "encoding"
2. If you are searching a text field and the character is on the list of reserved characters or words, precede them with two backslashes \\

Term modifiers

Wildcard search	Fuzzy search
Replace single character with ? e.g. te?t	Add ~ to the end of a single term e.g. roam~
Replace multiple characters with e.g. win	
Proximity search	Boost term
Add ~ and a number to the end of a phrase in quotes e.g. text ~ "Atlassian jira"~10'	Add with a boost factor (a number) to the end of a search term e.g. atlassian4 jira

Word stemming

Field	Add ~ to the beginning of a single term e.g. ~customize
-------	--

