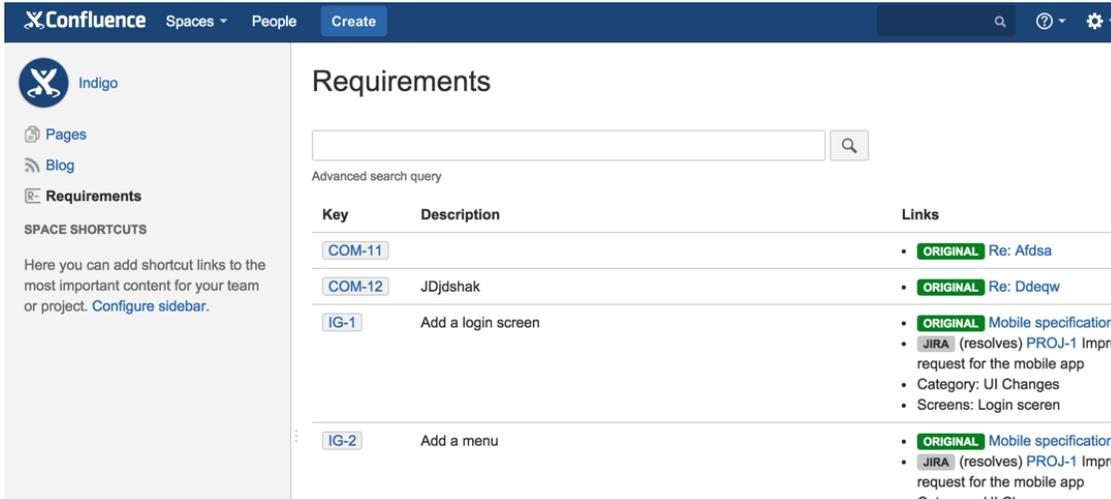


Search Syntax

This page describes the search syntax to find requirements in Confluence. Requirement Yogi provides a search screen in every space.



By default, only active requirements of the current space are displayed. Use the URL to display deleted requirements or to search across spaces.

Examples

Note: All those examples require Requirement Yogi 1.3.

Search on...	Query	Result
Key	key = 'IG-1'	The requirement with the exact key.
	key ~ 'IG-%'	All requirements starting with 'IG-'.
Contents	text ~ '% something'	All requirements whose text ends with 'something'.
Pages	page = 467382	All requirements defined on the page with ID 467382 (Does not include dependencies) (Before RY 3.1, it would retrieve all current-or-baseline requirements on all versions of this page). (Before RY 2.5, quotes around the page ID were necessary)
	page ~ 467382	Same as `links ~`, except it is not intuitive. (Before RY 3.1, it would retrieve all current-or-baselined requirements on all versions of that page).
	pageHistory ~ 123	All requirements on all versions of that page (Since RY 3.1)
	links = ...	All requirements defined or linked on the page.
Issues	jira = 'JRA-21'	All requirements linked from the issue JRA-21.
	jira@implements = 'JRA-21'	All requirements linked from the issue JRA-21 with the relationship "Implements" (since version 1.5)
	NOT (jira ~ '%')	Requirements which are not linked to any JIRA issue.
Properties	@Category = 'Functional'	All requirements where the property 'Category' = 'Functional'.
	ext@Category = 'Functional'	All requirements where the <i>external</i> property 'Category' = 'Functional'.
Dependencies	TO = 'REQ-001'	Requirements which reference REQ-001.
	FROM = 'REQ-001'	Requirements which are referenced by REQ-001.
	FROM ~ 'REQ-%'	Requirements which are referenced by any requirement starting with "REQ-".
	FROM@refines = 'REQ-001'	Requirements which are "refined" by REQ-001.

Special	isModified('7')	Requirements which have been modified since the baseline '7' of the current space. It works both with the baseline number and with the baseline name.
Baselines	baseline = 3	Requirements in their version 3.
	baseline was 3	Requirements in their current version, which were also in the baseline 3 (since 3.2.0).
	baseline = 4 and baseline was 3	Requirements in their version 4, which also where in baseline 3 (since 3.2.0).
	baseline = 'My Baseline'	Assuming 'My Baseline' exists in this space and has the number 3, it is equivalent to the previous query.
Tests	hasLastTest('%Success%')	Checks the last test is equal to the provided value, using '%' as a wildcard. Syntax: hasTest([relationship,] expectedResult [,page]) Syntax: hasLastTest([relationship,] expectedResult [,page])
Excel	excel = '48496653'	All requirements imported from the Excel attachment with ID 48496653 (since version 2.0)
	excel ~ '%'	All requirements imported from an Excel file.

Keywords

Keyword	Function
AND, OR, NOT	Boolean operators
= '...'	Strict equality.
== '...'	
~ '...%...'	Soft equality. Use % in the string as a wildcard.
@...	Reference to a property of the requirement (if you've defined requirements in columns).
IS NOT NULL	Can be used for the field 'baseline'

Fields

Field	Definition
key = ...	The key of the requirement. Keys are unique per space.
space Key = ...	The space key of the requirement (case sensitive).
baseline = ...	The baseline name or number. All statuses are implicitly included. Quotes were required before version 2.4: baseline = '1' Starting from 2.6.3, it also supports a variable: \$currentBaseline. The traceability matrix will display a field where you can input the number, and the saved matrix can be used in the blueprints of the baselines.
baseline was ...	The baseline name or number of a previous version of this requirement (since 3.2.0).
status = ...	The status of the requirement (ACTIVE, DELETED, MOVED). Default: ACTIVE. There is generally no reason to use it, since those statuses are internal details about requirements. You would rather define your own property named "Status", and search them using @status = 'Approved' for example. Do not confuse "status" (our internal implementation detail) and "@status" (any user-defined property).
text = ...	The contents of the requirement (Does not include the properties).
page = ...	The ID of a page or a version where a requirement is defined.
pageHistory = ...	The ID of a current page, where a requirement is defined. "~" is undefined.

links = ...	The ID of a page where a requirement is linked (or defined). "~" is undefined.
jira = ...	A JIRA issue linked to the requirement.
jira@re lations hip = ...	A JIRA issue linked with a specific relationship.
@apro perty = ...	A property with name "aproerty".
ext@a propert y = ...	An external property with name "aproerty".
to@are lations hip = ... from@ arelatio nship = ...	A dependency with the relationship "arelationship".
excel = ...	The ID of the attachment were those requirements are defined.

Changes in 3.2

Added 'baseline was'.

Changes in 3.1

The "page ~" and "page =" was restricted to requirements exactly on this page, instead of requirements on any version of this page.

"pageHistory" and "links" were added.

Formatting, lists of properties, etc.

Since [Release 2.4 - Improvements to the search](#), we support emoticons such as    etc. ! We also support lists! How does it work?

- We save the rendered format in HTML,
- We save the searchable format separately.

Thanks to this, you can search with:

- @property = '(/)' Searches for the  emoticon
- @property = 'item1' Searches for a property which has the item "item1" in the list. Note that the list may also have "item2", "item3", etc.
- @property = user('admin') Searches for the mention of the user 'admin' or the user key 'admin' (the second one will be stable over time, even if the username changes).

What about JQL in Jira?

See [JQL Syntax](#).