linkedTo()

This JQL function returns issues that are **linked** to any issue that is returned by a **JQL subquery**.

Additionally you can further refine your query by specifying **distinct link types** and even a **minimum number of linked issues**.

Syntax
`linkedTo(subquery) #Output: Issue list`

Examples

Parser expression	Description
`issue in linkedTo('project = "Asset Management" and type = Employee')`	This example returns all issues linked to the **Employees** tracked in the **Asset Management** project.
`issue in linkedTo('issuetype = Task and priority = Blocker')`	This example returns all issues that are linked to a **Task** with **priority** set to **Blocker.**

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
subquery	TEXT	A **JQL query** to select the issues that are linked to the those issues you want to be returned. **Pro tip** Enclose the subquery with single quotes **'** instead of double quotes **""**. Otherwise, every time a **double-quotation** is used in the subquery, they must be jumped with a **slash** /.

Output

This function returns an ISSUE LIST

Variant where you can additionally limit the results by specifying certain **link types.**

```
linkedTo(subquery, linkTypes) #Output: Issue list
```

Examples

Parser expression	Description
`issue in linkedTo('project = "Asset Management" and type = "Employee"","has asset")`	This example returns all issue linked to the **Employees** tracked in the **Asset Management** project via the "**has asset**" link type.
`issue in linkedTo('issuetype = Task and priority = "Blocker"', "is blocked by, causes")`	This example returns all issues that are linked to a **Task** with **priority** set to **Blocker** either by the "**is blocked by**" or "**causes**" link type .

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
subquery	TEXT	A **JQL query** to select the issues that are linked to the those issues you want to be returned. **Pro tip** Enclose the subquery with single quotes **''** instead of double quotes **""**. Otherwise, every time a **double-quotation** is used in the subquery, they must be jumped with a **slash** /.
linkTypes	TEXT	The **link type** or **link types** that must have been used to link to the desired target issues. The list of link types should be a **comma-separated list**.

Output

This function returns an ISSUE LIST

Variant where you can additionally specify a **minimum number** of linked issues.

Syntax

```
linkedTo(subquery, linkTypes, minimumLinked) #Output: Issue list
```

Examples

Parser expression	Description
```issue in linkedTo('project = "Asset Management" and type = Employee',"has asset",2)```	This example returns all issue linked to the **Employees** tracked in the **Asset Management** project via the "**has asset**" link type. In order to be returned by the query each Employee must have **at least two** of those linked issues.
```issue in linkedTo('project = "Asset Management" and type = PC',"belongs to",2) AND type = Employee```	This example returns all **Employees** that have a **PC** associated within the **Asset Management** project, that is also associated with another person.

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
`subquery`	TEXT	A **JQL query** to select the issues that are linked to the those issues you want to be returned. **Pro tip** Enclose the subquery with single quotes **'** instead of double quotes **""**. Otherwise, every time a **double-quotation** is used in the subquery, they must be jumped with a **slash** /.
`linkTypes`	TEXT	The **link type** or **link types** that must have been used to link to the desired target issues. The list of link types should be a **comma-separated list**.
`minimumLinked`	NUMBER	A **natural** number **larger** than **0**.

Output

This function returns an ISSUE LIST

The result is **not what you expected**? The number of returned issues feels **too low**?

By default, the **maximum number** of issues that will be returned by the JQL subquery, and thus can be processed by the logical expression is **1 000**.

To ensure the performance of your entire Jira instance, we limit the execution of the JQL function in terms of **issue count** and **execution time**. Please refer to your admin if you need to change the configuration.

Use cases and examples

Use case	JQL function
List all issues that are blocking issues that are due soon	linkedTo()