usersWithTextProperty()

This JQL function returns users by checking against their **user properties** that contain **text values**.

Syntax

```
usersWithTextProperty(property, operator, value) #Output: Text list
```

Examples

Parser expression	Description
`assignee in usersWithTextProperty ("country" , "=", "Germany")`	This example returns all issues where the current assignee has a **text property** called **country** with a value **equal to Germany**.
`reporter in usersWithTextProperty ("country", "~", "usa")`	This example returns all issues where the current reporter has a user **text reporter** called **country** with a value **equal to usa** ignoring the case. So even if the property was set to "USA" or "Usa", the function would count this as valid property value.
`reporter in usersWithTextProperty ("department", "matches", "(IT|HR)")`	This example returns all issues where the current **reporter** has a **text property** called **department** with a value **equal to IT** or **HR**. When using the **matches** operator, the provided **value** must be a valid **regex**.

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
property	TEXT	The name of the user property.
operator	TEXT	The operator used to compare the actual value with the expected value. **Allowed operators are**:

Comparator	Description
=	Property must be equal to the specified value.
!=	Property must not be equal to the specified value.
~	Property must be equal to the specified value, ignoring the case.
!~	Property must not be equal to the specified value, ignoring the case.
matches	A **regex** can be provided that must be fulfilled by the property.
doesn't match	A **regex** can be provided that must not be fulfilled by the property.

| value | TEXT | The value that will be used to check against the actual user property.

In case the **operator** is "matches" or "doesn't match", this value must be a **regex**. |
|-------|------|---------|

Output

This function returns a TEXT LIST

If you want to search for issues where a user as a specific **numeric or date property value**, have a look the the usersWithNumericProperty() or usersWithDateTimeProperty() function.

If you want to search for issues, checking only if a **property is being set ignoring it's value**, have a look the the usersWithProperty() function.

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.