

getMatchingValue()

Supported list types

[Number list](#) | [Text list](#) | [Use cases and examples](#)

1 2 3 Number list

This function returns the **numeric reference value** or a given text. If the **key** is not in the **keyList**, the function returns

NULL

Syntax

getMatchingValue(key, keyList, referenceList) #Output: Number

Examples

Parser expression	Description
<pre>getMatchingValue("Three", ["One", "Two", "Three", "Four", "Five"], [1, 1+1, 3*1, 4, 4+1])</pre>	This example returns: 3
<pre>getMatchingValue("Critical", ["Normal", "High", "Critical"], [1, 5, 20])</pre>	This example returns: 20
<pre>%{getMatchingValue("None", ["Normal", "High", "Critical"], [1, 5, 20, 0])}</pre>	This example returns 0 The textKey "None" is not available in the textList , so the last value of the referenceNumberList is returned (it has one more value than the textList).
<pre>getMatchingValue(%{issue.priority}, ["High", "Medium", "Low"], [3, 2, 1])</pre>	This example returns the correct number depending on the current priority . This example could be used e.g. in the Update or copy field values post function to set a specific field depending on the current priority.

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
key	TEXT	A text that exists in the keyList .
keyList	TEXT LIST	Any given text list.
referenceList	NUMBER LIST	Any given number list.

Output

This function returns a

NUMBER

Variant with a **key** and two **number lists**.

Syntax

```
getMatchingValue(key, keyList, referenceList) #Output: Number
```

Examples

Parser expression	Description
<pre>getMatchingValue(5, [1, 3, 5, 7, 9], [1, 1+1, 3*1, 4, 4+1])</pre>	This example returns: 3
<pre>%{getMatchingValue(3, [1, 5, 7, 9], [1, 2, 3, 4, 99])}</pre>	This example returns: 99 The numberKey 3 is not available in the numberList , so the last value of the referenceNumberList is returned (it has one more value than the numberList).

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
key	NUMBER	A number that exists in the keyList .
keyList	NUMBER LIST	Any given number list.
referenceList	NUMBER LIST	Any given number list.

Output

This function returns a

NUMBER



Text list

This function returns the **text reference value** or a given text.

Syntax

```
getMatchingValue(key, keyList, referenceList) #Output: Text
```

Examples

Parser expression	Description
<pre>getMatchingValue("Spain", ["USA", "UK", "France", "Spain", "Germany"], ["Washington", "London", "Paris", "Madrid", "Berlin"])</pre>	This example returns: "Madrid"
<pre>getMatchingValue(%{issue.issueType}, ["Task", "Sub-Task", "Story"], ["This is the description for a Task.", "Description for a Sub-Task", "Here comes the Story."])</pre>	This example returns the correct description depending on the current issue type. This could be used e.g. in the Create issue or Update or copy field values post function to set the description depending on the selected issue type.

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
key	TEXT	A text that exists in the keyList
keyList	TEXT LIST	Any given text list
referenceList	TEXT LIST	Any given text list

Output

This function returns a

Variant with a **number list**.

Syntax

```
getMatchingValue(key, keyList, referenceList) #Output: Text
```

Examples

Parser expression	Description
<pre>getMatchingValue(8, [2, 4, 6, 8, 10], ["Washington", "London", "Paris", "Madrid", "Berlin"])</pre>	This example returns: "Madrid"

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
key	NUMBER	A number that exists in the keyList
keyList	NUMBER LIST	Any given number list
referenceList	TEXT LIST	Any given text list

Output

This function returns a

Use cases and examples

Use case	JWT feature	Workflow function	Field type	Automated action	Parser functions
Create a sub-task for every user in a field and assign them		Create issue			getMatchingValue() toStringList()
Create a sub-task for every recurring deadline within a task		Create issue			addMonths() getMatchingValue() dateToString()
Add sub-tasks to an automatically created issue	 	Create issue		Create issue action	getMatchingValue()
Create three issues with individual summaries		Create issue			getMatchingValue()