

# Lists

## On this page

[Fixed values](#) | [Available operators](#) | [Order of operations](#) | [List functions for multiple list types](#) | [Number list \(only\) functions](#) | [Issue list \(only\) functions](#)

JWT is capable of processing various types of lists, including **text**, **number** and **issue lists**. This page contains valuable information about working with lists.

The **list data type** is an ordered list of elements. Those elements have a certain data type (text or number).

It is possible to:

- access individual elements (e.g. using the function [nthElement\(\)](#)),
- create lists out of [virtual fields](#) (e.g. using [toStringList\(\)](#)) or
- use the list functions presented on this page to work with lists.

## Fixed values

A **list** can be written in literal as a **comma separated list of texts in double quotes written inside brackets**.

| Expression                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>[ ]</pre> #All lists are enclosed in brackets.         | An <b>empty</b> list.                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <pre>["Blue", "Green", "Yellow", "Orange", "Red"]</pre>     | A <b>text</b> list with five elements.                                                                                                                                                                                                                                                                                                                                                                                                       |
| <pre>[1, 2, 3]</pre>                                        | A <b>number</b> list with three elements.                                                                                                                                                                                                                                                                                                                                                                                                    |
| <pre>[1, {issue.subtasks.count}, {issue.links.count}]</pre> | A <b>number</b> list with three elements using field codes and <a href="#">JWT expression parser functions</a> .<br><br>To achieve this, the following virtual field codes: <ul style="list-style-type: none"><li>• <a href="#">Number of sub-tasks</a></li><li>• <a href="#">Number of linked issues</a></li></ul>                                                                                                                          |
| <pre>toStringList(%<br/>{issue.components})</pre>           | A text list with all <b>components</b> of the <b>current issue</b> .<br><br><div>Fields codes of multi-valued fields, such as components, labels, etc. by default return a text with a <b>comma separated list of values</b>. If you want to individually process these elements, you need to convert the text into a text list using <a href="#">toStringList()</a>.<br/><br/><a href="#">Learn more about Converting data types.</a></div> |

## Reading values from issue lists

Most of the times, lists are being defined or returned using field codes or [JWT expression parser functions](#). In order to read or retrieve values from an issue list, functions such as [fieldValue\(\)](#) must be used.

| Expression                                                                       | Description                                                                                                                                                                                                                                                                                                                                                     |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>fieldValue(%{issue.assignee}, subtasks())</code>                           | <p>A list of all <b>assignees</b> of the current <b>issues's sub-tasks</b>.</p> <p>The list may contain duplicate user names.</p> <p>To achieve this, the following functions were used:</p> <ul style="list-style-type: none"> <li>• <code>fieldValue()</code></li> <li>• <code>subtasks()</code></li> </ul>                                                   |
| <code>distinct(fieldValue(%{issue.assignee}, subtasks()))</code>                 | <p>A list of all <b>distinct assignees</b> of the current <b>issues's sub-tasks</b>.</p> <p>The list only contains <b>unique</b> user names.</p> <p>To achieve this, the following functions were used:</p> <ul style="list-style-type: none"> <li>• <code>distinct()</code></li> <li>• <code>fieldValue()</code></li> <li>• <code>subtasks()</code></li> </ul> |
| <code>fieldValue(%{issue.priority}, linkedIssues("blocks, is cloned by"))</code> | <p>A list of priorities of those issues linked to current issue through issue link types "<b>blocks</b>" and "<b>is cloned by</b>".</p> <p>To achieve this, the following functions were used:</p> <ul style="list-style-type: none"> <li>• <code>fieldValue()</code></li> <li>• <code>linkedIssues()</code></li> </ul>                                         |
| <code>distinct(fieldValue(%{issue.components}, issuesUnderEpic()))</code>        | <p>A list with <b>distinct components</b> of all <b>issues</b> which are <b>linked to same epic as the current issue</b>.</p> <p>To achieve this, the following functions were used:</p> <ul style="list-style-type: none"> <li>• <code>distinct()</code></li> <li>• <code>fieldValue()</code></li> <li>• <code>issuesUnderEpic()</code></li> </ul>             |

JWT offers individual operators that can be used when working with Lists.



## Available operators

| Function               | Short description                                  | Output |
|------------------------|----------------------------------------------------|--------|
| <code>APPEND</code>    | <b>Combines</b> the elements of two <b>lists</b> . | LIST   |
| <code>UNION</code>     | Returns <b>distinct elements</b> of two lists.     | LIST   |
| <code>INTERSECT</code> | Returns <b>common elements</b> of two lists.       | LIST   |
| <code>EXCEPT</code>    | <b>Removes</b> certain elements from a list.       | LIST   |



## Order of operations

If you use multiple operators in a single expression, they will follow a certain order in which they are processed or a precedence.

| OPERATORS              | PRECEDENCE  | ASSOCIATIVITY |
|------------------------|-------------|---------------|
| <code>INTERSECT</code> | 1 (highest) | Left-to-right |

|                       |            |               |
|-----------------------|------------|---------------|
| APPEND, EXCEPT, UNION | 2 (lowest) | Left-to-right |
|-----------------------|------------|---------------|

- When using the list operators, you have to make sure that both lists that you compare are of the **same type**.
- All operators are **case insensitive**, i.e., they can also be written in lower case: **append**, **union**, **intersect** and **except**.
- There are **four equivalent functions** available for each type of list, and their behavior is **exactly equivalent** to that of its corresponding operator.
  - `append()`
  - `except()`
  - `intersect()`
  - `union()`
- This way, you can choose to use **operators** or **functions** according to your preference. Although operators yield shorter expressions and with fewer parentheses, the usage of functions produces a more functional consistent syntax.

## List functions for multiple list types

The following list contains all the available functions that work with all kinds of lists; text, [number](#) and [issue](#) lists.

| Function                           | Short description                                                                            | Output                                                                                                              | Label                      |
|------------------------------------|----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|----------------------------|
| <code>append()</code>              | <b>Combines</b> the elements of two <b>lists</b> .                                           | <input type="text" value="LIST"/>                                                                                   |                            |
| <code>count()</code>               | Returns the <b>number of elements</b> in a text, number or issue list.                       | <input type="text" value="NUMBER"/>                                                                                 | <a href="#">STAFF PICK</a> |
| <code>distinct()</code>            | <b>Removes all duplicates</b> from a number, text, or issue list.                            | <input type="text" value="LIST"/>                                                                                   | <a href="#">STAFF PICK</a> |
| <code>except()</code>              | <b>Removes</b> certain elements from a list.                                                 | <input type="text" value="LIST"/>                                                                                   |                            |
| <code>filterByCardinality()</code> | Filters a given number, text, or issue list by the <b>number of occurrence of elements</b> . | <input type="text" value="LIST"/>                                                                                   |                            |
| <code>filterByPredicate()</code>   | Filters a number, issue, or a text list by a given <b>logical expression</b> .               | <input type="text" value="LIST"/>                                                                                   | <a href="#">STAFF PICK</a> |
| <code>filterByValue()</code>       | <b>Filters</b> a number or text list using a <b>given comparison</b> .                       | <input type="text" value="LIST"/>                                                                                   | <a href="#">STAFF PICK</a> |
| <code>first()</code>               | Returns the first element of a <b>number</b> , <b>text</b> , or <b>issue list</b> .          | <input type="text" value="NUMBER"/><br><input type="text" value="TEXT"/><br><input type="text" value="ISSUE LIST"/> | <a href="#">STAFF PICK</a> |
| <code>getMatchingValue()</code>    | Returns a custom reference value for a given or <b>text</b> or <b>number</b> .               | <input type="text" value="NUMBER"/><br><input type="text" value="TEXT"/>                                            |                            |
| <code>getRemoteLinks()</code>      | Returns a list of remote links.                                                              | <input type="text" value="TEXT LIST"/>                                                                              |                            |
| <code>indexOf()</code>             | Returns the <b>index / position</b> of a <b>specific element</b> in a list.                  | <input type="text" value="NUMBER"/>                                                                                 |                            |
| <code>intersect()</code>           | Returns <b>common elements</b> of two lists.                                                 | <input type="text" value="LIST"/>                                                                                   |                            |
| <code>invertList()</code>          | Inverts the <b>order</b> of a given list.                                                    | <input type="text" value="LIST"/>                                                                                   |                            |
| <code>last()</code>                | Returns the last element of a <b>number</b> , <b>text</b> , or <b>issue list</b> .           | <input type="text" value="NUMBER"/><br><input type="text" value="TEXT"/><br><input type="text" value="ISSUE LIST"/> |                            |

|                                 |                                                                                                                                             |                              |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| <code>mathOnStringList()</code> | Returns a <b>number list</b> with results of the given calculation performed for each text in the specified list.                           | NUMBER LIST                  |
| <code>nthElement()</code>       | Returns the <b>nth element</b> of a number, string or issue list.                                                                           | NUMBER<br>TEXT<br>ISSUE LIST |
| <code>sort()</code>             | <b>Sorts</b> a given list in a specific order.                                                                                              | LIST                         |
| <code>sublist()</code>          | Returns a <b>defined extract</b> of a given list.                                                                                           | LIST                         |
| <code>textOnNumberList()</code> | Returns a <b>text list</b> in a result of evaluating <b>text_expression</b> against each of the numeric values in argument <b>numbers</b> . | TEXT LIST                    |
| <code>textOnNumberList()</code> |                                                                                                                                             |                              |
| <code>textOnStringList()</code> | Matches elements of a text list against a <b>text expression</b> .                                                                          | TEXT LIST                    |
| <code>union()</code>            | Returns <b>distinct elements</b> of two lists.                                                                                              | LIST                         |

## Number list (only) functions

The following list contains all the available functions that work with **number lists only**.

| Function                        | Short description                                                                                                   | Output      | Label      |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------|-------------|------------|
| <code>avg()</code>              | Calculates the <b>average</b> values of a given number list.                                                        | NUMBER      |            |
| <code>mathOnNumberList()</code> | Returns a <b>number list</b> with results of the given calculation performed for each number in the specified list. | NUMBER LIST |            |
| <code>max(list)</code>          | Returns the <b>highest value</b> in a number list.                                                                  | NUMBER      |            |
| <code>min(list)</code>          | Returns the <b>smallest value</b> in a number list.                                                                 | NUMBER      |            |
| <code>sum()</code>              | <b>Sums up</b> all values in a given number list.                                                                   | NUMBER      | STAFF PICK |

## Issue list (only) functions

The following list contains all the available functions that work with **issue lists only**.

| Function                          | Short description                                                                                                       | Output                   | Label      |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------|--------------------------|------------|
| <code>allIssuesUnder()</code>     | Returns issues located in <b>any level under</b> a given parent issue according to <b>Advanced Roadmaps hierarchy</b> . | ISSUE LIST               | STAFF PICK |
| <code>epic()</code>               | Returns <b>all epics</b> linked to specified issues.                                                                    | ISSUE LIST               |            |
| <code>fieldValue()</code>         | Returns a number or text list with <b>field values</b> .                                                                | NUMBER LIST<br>TEXT LIST | STAFF PICK |
| <code>filterByFieldValue()</code> | <b>Filters</b> an <b>issue list</b> using a given comparison for field values.                                          | ISSUE LIST               |            |
| <code>filterByIssueType()</code>  | <b>Filters</b> a given issue list by <b>issue type</b> .                                                                | ISSUE LIST               |            |

|                                         |                                                                                                                                     |             |            |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|-------------|------------|
| <code>filterByProject()</code>          | <b>Filters</b> a given issue list by <b>project</b> .                                                                               | ISSUE LIST  |            |
| <code>filterByProjectCategory()</code>  | <b>Filters</b> a given issue list by <b>project category</b> .                                                                      | ISSUE LIST  |            |
| <code>filterByResolution()</code>       | <b>Filters</b> a given issue list by <b>resolution</b> .                                                                            | ISSUE LIST  |            |
| <code>filterByStatus()</code>           | <b>Filters</b> a given issue list by <b>issue status</b> .                                                                          | ISSUE LIST  | STAFF PICK |
| <code>filterByStatusCategory()</code>   | <b>Filters</b> a given issue list by <b>status category</b> .                                                                       | ISSUE LIST  |            |
| <code>getIssuesFromProjects()</code>    | Returns <b>all issues</b> from specified projects.                                                                                  | ISSUE LIST  |            |
| <code>issuesAbove()</code>              | Returns all issues located in <b>any level above</b> a given parent issue according to <b>Advanced Roadmaps hierarchy</b> .         | ISSUE LIST  | STAFF PICK |
| <code>issuesFromJQL()</code>            | Returns a list of issues returned by a specified <b>JQL query</b> .                                                                 | ISSUE LIST  | STAFF PICK |
| <code>issuesUnder()</code>              | Returns issues located in the level <b>just under</b> a given parent issue according to <b>Advanced Roadmaps hierarchy</b> .        | ISSUE LIST  | STAFF PICK |
| <code>issuesUnderEpic()</code>          | Returns <b>all issues</b> linked to a given <b>epic</b>                                                                             | ISSUE LIST  | STAFF PICK |
| <code>linkedIssues()</code>             | Returns a <b>list of issues linked</b> .                                                                                            | ISSUE LIST  | STAFF PICK |
| <code>mathOnIssueList()</code>          | Returns a <b>number list</b> with results of the given calculation performed for each issue in the specified list.                  | NUMBER LIST |            |
| <code>numberOfRemoteIssueLinks()</code> | Returns the number of <b>issue links</b> to remote Jira instances                                                                   | NUMBER      |            |
| <code>parent()</code>                   | Returns the <b>direct parent(s)</b> of the given issue(s) according to <b>Advanced Roadmaps hierarchy</b> .                         | ISSUE LIST  | STAFF PICK |
| <code>siblingIssues()</code>            | Returns all issues which are located <b>directly under a given issue's parent</b> according to <b>Advanced Roadmaps hierarchy</b> . | ISSUE LIST  | STAFF PICK |
| <code>siblingIssuesUnderEpic()</code>   | Returns all <b>sibling issues</b> linked to the same <b>epic</b>                                                                    | ISSUE LIST  |            |
| <code>siblingSubtasks()</code>          | Returns all <b>sibling sub-tasks</b> .                                                                                              | ISSUE LIST  |            |
| <code>subtasks()</code>                 | Returns <b>sub-tasks</b> of given issues.                                                                                           | ISSUE LIST  | STAFF PICK |
| <code>textOnIssueList()</code>          | Returns a <b>text list</b> in result of evaluating <b>textExpression</b> against each of the issues in argument <b>issues</b> .     | TEXT LIST   |            |
| <code>transitionLinkedIssues()</code>   | Returns a list of all <b>issues linked</b> during the <b>transition</b> .                                                           | ISSUE LIST  |            |
| <code>transitivelyLinkedIssues()</code> | Returns issues that are <b>directly</b> or <b>transitively linked</b> to the current issue.                                         | ISSUE LIST  |            |

If you still have questions, feel free to refer to our [support](#) team.