

# toString()

This function converts the decimal representation of a **number** (or timestamp) **into** plain **text**.

This is one of the **most commonly used** functions in JWT.

Whenever any type of **list** is returned by a **field code** or parser function (e.g. `subtasks()`) and you are using the **advanced text parsing mode**, the **toString()** function converts the output to a **readable** format.

This is especially helpful if you want to **test your expression**.

### Syntax

```
toString(number) #Output: Text
```

## Examples

| Parser expression               | Description   |
|---------------------------------|---|
| <code>toString(3.141592)</code> | This example returns the text:<br><b>"3.141592"</b> |

## Additional information

Parameters used in this function

| Parameter | Input (data type) | Description                               |
|-----------|-------------------|---|
| number    | <div>NUMBER</div> | Any given number or numerical field code. |

## Output

This function returns a 

TEXT

Variant where you can additionally define the number of **decimal places**.

### Syntax

```
toString(number, decimalPlaces) #Output: Text
```

## Examples

| Parser expression                | Description                                     |
|----------------------------------|---|
| <pre>toString(3.141592, 2)</pre> | This example returns the text:<br><b>"3.14"</b> |

## Additional information

Parameters used in this function

| Parameter     | Input (data type) | Description                 |
|---------------|-------------------|-----------------------------|
| number        | NUMBER            | Any given number.           |
| decimalPlaces | NUMBER            | Any <b>positive</b> number. |

## Output

This function returns a TEXT

## 1 2 3 Number list

This function converts any **number list into plain text**.

A comma-separated text with the decimal representation of each numeric value will be returned.

### Syntax

```
toString(numberList) #Output: Text
```

## Examples

| Parser expression                   | Description   |
|-------------------------------------|---|
| <pre>toString([1, 2, 3, 4.0])</pre> | This example returns the text:<br><b>"1, 2, 3, 4"</b> |

## Additional information

Parameters used in this function

| Parameter  | Input (data type) | Description            |
|------------|-------------------|------------------------|
| numberList | NUMBER LIST       | Any given number list. |

## Output

This function returns a TEXT

Variant where you can additionally define the number of **decimal places**.

### Syntax

```
toString(numberList, decimalPlaces) #Output: Text
```

## Examples

| Parser expression                               | Description   |
|---|---|
| <pre>toString([1.123, 2.452, 3.64612], 2)</pre> | This example returns the text:<br><b>"1.12, 2.45, 3.65"</b> |

## Additional information

Parameters used in this function

| Parameter     | Input (data type)        | Description            |
|---------------|--------------------------|------------------------|
| numberList    | <span>NUMBER LIST</span> | Any given number list. |
| decimalPlaces | <span>NUMBER</span>      | Any positive number.   |

## Output

This function returns a TEXT

Variant where you can additionally define a specific **separator**.

### Syntax

```
toString(numberList, decimalPlaces, separator) #Output: Text
```

## Examples

| Parser expression | Description |
|-------------------|-------------|
|-------------------|-------------|

```
toString([1.123, 2.452, 3.64612], 2, " : ")
```

This example returns the text:

**"1.12 : 2.45 : 3.65"**

## Additional information

Parameters used in this function

| Parameter     | Input (data type) | Description              |
|---------------|-------------------|--------------------------|
| numberList    | NUMBER LIST       | Any given number list.   |
| decimalPlaces | NUMBER            | Any positive number.     |
| separator     | TEXT              | Any separator character. |

## Output

This function returns a TEXT



### Text list

This function converts any **text list into plain text**.

A comma-separated text with each value will be returned.

#### Syntax

```
toString(textList) #Output: Text
```

## Examples

| Parser expression  | Description  |
|--|--|
| <pre>toString(["Hello", " ", "world", "!"])</pre>              | This example returns the text:<br><b>"Hello, , world, !"</b>   |
| <pre>toString(fieldValue(%{issue.reporter}, subtasks()))</pre> | This example returns the user names of all <b>sub-task reporters</b> , e.g.:<br><b>"b.smith, a.grant"</b><br><br>To achieve this the following functions are used: <ul style="list-style-type: none"><li><code>fieldValue()</code></li><li><code>subtasks()</code></li></ul> |

## Additional information

Parameters used in this function

| Parameter | Input (data type)                      | Description          |
|-----------|--|----------------------|
| textList  | <input type="text" value="TEXT LIST"/> | Any given text list. |

## Output

This function returns a

Variant where you can additionally define a specific **separator**.

### Syntax

```
toString(textList, separator) #Output: Text
```

## Examples

| Parser expression                                   | Description   |
|---|---|
| <pre>toString(["blue", "red", "green"], "; ")</pre> | This example returns the text:<br><b>"blue; red; green"</b> |

## Additional information

Parameters used in this function

| Parameter | Input (data type)                      | Description              |
|-----------|--|--------------------------|
| textList  | <input type="text" value="TEXT LIST"/> | Any given text list.     |
| separator | <input type="text" value="TEXT"/>      | Any separator character. |

## Output

This function returns a



### Issue list

Converts an **issue list** into a comma-separated text containing **issue keys**.

### Syntax

```
toString(issueList) #Output: Text
```

## Examples

| Parser expression               | Description   |
|---------------------------------|---|
| <pre>toString(subtasks())</pre> | <p>This example might return:</p> <p><b>"CRM-5, CRM-6 "</b></p> <p>To achieve this the following functions are used:</p> <ul style="list-style-type: none"><li>• <a href="#">subtasks()</a></li></ul> |

## Additional information

Parameters used in this function

| Parameter | Input (data type)                       | Description   |
|-----------|---|---|
| issueList | <input type="text" value="ISSUE LIST"/> | Any given issue list. Usually this value is retrieved from a function (e.g. <a href="#">linkedIssues()</a> or <a href="#">subtasks()</a> ). |

## Output

This function returns a

Variant where you can additionally define a specific **separator**.

### Syntax

```
toString(issueList, separator) #Output: Text
```

## Examples

| Parser expression                    | Description   |
|--------------------------------------|---|
| <pre>toString(subtasks(), " ")</pre> | <p>This example might return:</p> <p><b>"CRM-5 CRM-6 CRM-7"</b></p> |
| <pre>toString(subtasks(), ";")</pre> | <p>This example might return:</p> <p><b>"CRM-5;CRM-6;CRM-7"</b></p> |

## Additional information

Parameters used in this function

| Parameter | Input (data type) | Description   |
|-----------|-------------------|---|
| issueList | ISSUE LIST        | Any given issue list. Usually this value is retrieved from a function (e.g. <code>linkedIssues()</code> or <code>subtasks()</code> ). |
| separator | TEXT              | Any separator character.  |

## Output

This function returns a

TEXT



## Use cases and examples

| Use case   | JWT feature | Workflow function           | Field type | Automated action                                  | Parser functions   |
|--|-------------|-----------------------------|------------|---|--|
| Add comment with the request participants' display name                |             | Add comment                 |            |   | <code>toString()</code> <code>textOnStringList()</code> <code>toStringList()</code> <code>userDisplayName()</code>   |
| Display reporters of linked Cloud issues                               |             |                             | Text       |   | <code>executeRemoteAction()</code><br><code>findPattern()</code><br><code>toStringList()</code><br><code>toString()</code><br><code>distinct()</code>                |
| Set fix version based on its start and release date                    |             | Update or copy field values |            |   | <code>toString()</code> <code>textOnStringList()</code> <code>unreleasedVersions()</code> <code>startDates()</code> <code>releaseDates()</code> <code>first()</code> |
| Obtain a value from an Elements Connect custom field                   |             | Send email                  |            |   | <code>findPattern()</code><br><code>toString()</code>  |
| Display historic due dates   |             |                             | Text       |   | <code>toString()</code><br><code>textOnNumberList()</code><br><code>fieldHistory()</code><br><code>dateToString()</code>   |
| Link to JQL query with information from issue                          |             |                             | Text       |   | <code>toString()</code><br><code>textOnStringList()</code><br><code>toStringList()</code>  |
| Send email with the URL of the attachments included in the description |             | Send email                  |            |   | <code>replaceAll()</code> <code>toString()</code> <code>attachmentUrls()</code> <code>findPattern()</code>   |
| Add links to the issues in a ScriptRunner Issue Picker field           |             |                             |            | Create issue link action Delete issue link action | <code>issueKeysToIssueList()</code> <code>toString()</code> <code>toStringList()</code> <code>replaceAll()</code> <code>previousValue()</code>                       |

|   |   |                             |                          |   |
|---|---|-----------------------------|--------------------------|---|
| Add components of the epic to the current issue     |    |                             | Update field action      |   |
| Link issue to issue keys in its description         |    | Create issue link           |                          | findPattern() toString()  |
| Create issue under epic                             |    | Create issue                |                          | toString() epic()   |
| Prevent external users from creating issues         |    | Logical validator           |                          | matches() toString() textOnStringList() toStringList() userDisplayName()  |
| Create a sub-task for every sub-task closed         |    | Create issue                |                          | findReplaceFirst() first() findPattern() toString() sum() toNumber()  |
| Add watchers ignoring inactive users                |    | Update or copy field values |                          | usersInGroup() isActive() toString() filterByPredicate()  |
| Link to the linked epic from the current issue      |    |                             | Text                     | toString()<br>fieldValue()<br>issueKeysToIssueList()  |
| Clone epic, tasks and sub-tasks                     |    | Create issue                |                          | toString() toStringList() issueKeysToIssueList() replaceFirst() first() fieldValue() textOnStringList() findModify() subtasks() issuesUnderEpic() |
| Add a table with the elements of a text to an email |  | Send email                  |                          | toString() textOnStringList() toStringList()  |
| Match several values of a list                      |  | Update or copy field values |                          | toString() distinct() filterByPredicate()   |
| Track issues mentioned in comments                  |  |                             | Create issue link action | findPattern()<br>issueKeysToIssueList()<br>toString()   |
| Components from all sub-tasks                       |  |                             | Text                     | toString()<br>distinct()<br>fieldValue()<br>subtasks()  |
| Triage issues created by email                      |  | Move issue                  |                          | toUpperCase()<br>toString()<br>findPattern()<br>isInGroup()   |