

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: "3.141592"

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
number	NUMBER	Any given number or numerical field code.

Output

This function returns a

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

Syntax

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

Examples

Parser expression	Description
toString(3.141592, 2)	This example returns the text: "3.14"

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
number	NUMBER	Any given number.
decimalPlaces	NUMBER	Any positive number.

Output

This function returns a

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
toString([1, 2, 3, 4.0])	This example returns the text: "1, 2, 3, 4"

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
<code>toString([1.123, 2.452, 3.64612], 2)</code>	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.

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 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: "Hello, , world, !"
<pre>toString(fieldValue(%{issue.reporter}, subtasks()))</pre>	This example returns the user names of all sub-task reporters , e.g.: "b.smith, a.grant" To achieve this the following functions are used: <code>fieldValue()</code> <code>subtasks()</code>

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
textList	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: "blue; red; green"

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
textList	TEXT LIST	Any given text list.
separator	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>	This example might return: "CRM-5, CRM-6 " To achieve this the following functions are used: • subtasks()

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. linkedIssues() or subtasks()).

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>	This example might return: "CRM-5 CRM-6 CRM-7"
<pre>toString(subtasks(), ";")</pre>	This example might return: "CRM-5;CRM-6;CRM-7"

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



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> <code>findPattern()</code> <code>toStringList()</code> <code>toString()</code> <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> <code>toString()</code>
Display historic due dates			Text		<code>toString()</code> <code>textOnNumberList()</code> <code>fieldHistory()</code> <code>dateToString()</code>
Link to JQL query with information from issue			Text		<code>toString()</code> <code>textOnStringList()</code> <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() fieldValue() 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() issueKeysToIssueList() toString()
Components from all sub-tasks			Text	toString() distinct() fieldValue() subtasks()
Triage issues created by email		Move issue		toUpperCase() toString() findPattern() isInGroup()