

Data types

Throughout the documentation we refer to **data types** that can be used in the **expression parser** and its functions and field codes.

Data types define what kind of data a function needs as **input** and what you should expect as **output**.

Data types in JWT

Data type	Description
TEXT	This type represents any kind of text or character string including all kinds of select and multi-select fields
NUMBER	This type represents numeric values, and is also used to store date, time and date-time values. Learn more about Dates, times and time zones If you need the output as text, simply add % in front of the field code or directly insert the field code as text.
LIST	• Number list: This type represents a collection of numeric values returned by various functions. The size may vary from 0 to any number of numeric values. • Text list: This type represents a collection of text values returned by various functions. The size may vary from 0 to any number of text values. • Issue list: This type represents a collection of issues. The size may vary from 0 to any number of issues. It is returned by issue selection of filtering functions. Learn more about Lists If you need to "flatten" the list, or simply convert it to a text, to be written into any field, comment, etc., make sure to have a look at the toString() function!
BOOLEAN	Some operators, workflow functions, parser functions, and automation elements return a logical, or boolean, value of true or false. These examples include, but are not limited to: • Logical condition • Logical validator • Logical mode • Boolean condition

Converting (casting) data types

Converting (or **casting**) data types is very important throughout all functions provided by JWT.

To be able to **set**, **transform** or **calculate** values it might be necessary to turn a **text** value to a **number**, a **number** or a **timestamp** to a **text**, and even individual **text** or **number** values to a **list**.

The following functions will help you to achieve that.

[Available functions](#)



Function	Short description	Output
issueKeysToIssueList()	Converts a text containing issue keys into an issue list .	ISSUE LIST
toInteger()	Converts text into an integer .	NUMBER
toNumber()	Converts text into a number .	NUMBER
toNumberList()	Converts text containing numbers into a number list .	NUMBER LIST
toString()	Converts numbers and lists into plain text .	TEXT
toStringList()	Converts text into a text list .	TEXT LIST

Whenever you write a numeric term at the right-hand side of **the + operator** or a **comparison operator** like = , and the left-hand side is occupied by a text term, JWT will automatically transform the right-hand side term into a text [toString\(\)](#).

Operator	Example
+	"His age is " + 30 is equivalent to "His age is " + toString(30)
= (any comparison operator)	"30" = 30 is equivalent to "30" = toString(30)

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