

Test your expression

On this page

[The expression editor](#) | [The expression parser test page](#) | [5 steps to test your expression](#)

There are **two ways** to test expressions:

- the **Expression parser test page** (introduced with JWT 2.4.0)
- the built-in **expression preview** in the [JWT expression editor](#) (introduced with JWT 3.0.0)

The main purpose is always the same: get a preview of the outcome of an expression in order to deploy your configuration with confidence and without having to troubleshoot afterward.

The expression editor

When you want to test an expression directly in a JWT **workflow function** or **calculated field** configuration, you can instantly **test** and **preview** the **results** of in the [JWT expression editor](#).

The expression parser test page

The **Expression parser test page** is located **centrally** in the Jira administration: **Add-ons** **Jira Workflow Toolbox** **Expression parser test page**.

You want to test a logical expression or calculate a date? You simply want to know which **values** are currently stored in different issues? On the expression parser test page you can test **all kinds** of expressions since all [Parsing modes](#) are available.

Try adding a **field code** and play around with the different [Parsing modes](#) to preview the output!

5 steps to test your expression

1

Type in your **expression** or select one of the built-in **examples**.

The Expression parser test page does **not** support all [field codes](#). Field codes for transitional fields e.g. **Transition comment**, or the temporary fields are not supported.

2

Click on the **Run** button



3

Select an issue you want to test your expression with.

4

Confirm your selection by clicking on **Run**.

5

Check the **resulting output** or **analyze potential errors**.

Basic text
Add field
Examples

1 What is my parent? Ahhhh, it's `#{parent.key}`
[Line 1 / Col 4]

Enter plain text and optionally use [field codes](#), e.g. `%{issue.summary}`, to insert field values.

Test expression

Current issue picker

SCRUM-28 - This is a sibling sub-task

Select an issue as the current issue to preview the result of the expression.

Run
Reset
Close

✓ **Parsed expression**

What is my parent? Ahhhh, it's SCRUM-6

Syntax check

Before running your expression against a certain issue, it's recommended to check if the syntax is correct.

The **syntax check** button indicates, if the current expression in the input field is syntactically correct or not . The background check runs at least **one second after the last input**, but you can also click the **Syntax button** for enforce a check immediately.

Your browser does not support the HTML5 video element

The run button

Basic text
Add field
Examples

?

1 What is my parent? Ahhhh, it's `#{parent.key}`
[Line 1 / Col 0] Try your expression

By clicking the **Run** button, a test expression panel is displayed below your expression. To test your expression, do the following:

1. Select an **issue** that you want to test your expression with (as the current issue)
2. Click **Run** again

Your browser does not support the HTML5 video element

Error messages

Even if the syntax of the expression is correct, it may happen that the expression **result** is **erroneous**, e.g. when fields are empty.

In the example below, the **value** of the custom number field with id **12202** is returning a value which is not a valid parameter for the function `substring()`.

✓

fx Advanced text ▾

Add field ▾

Examples ▾

▶

?


1 substring(%{issue.summary}, 0, {issue.cf12202})

[Line 1 / Col 26] Try your expression

Enter plain text and optionally use **field codes**, e.g. %{issue.summary}, to insert field values.

Test expression ✕

Current issue picker


 TIS-131 - Customers reporting shopping cart purchasing issues with the TIS web store ✕ ▾

Select an issue as the current issue to preview the result of the expression.

Run

Reset

Close

 Error in call to "substring(s, beginIndex, endIndex)" function : String index out of range: -2

Output of different data types

When using the [JWT expression editor](#), the expressions can return different values and types.

Besides the expression entered, the selected parsing mode is mainly responsible for the returned value (and its type):

- [Basic text mode](#)
- [Advanced text mode](#)
- [Logical mode](#)
- [Numeric mode](#)
- [Text list mode](#)
- [Issue list mode](#)
- [JQL mode](#)
- [Mixed mode](#)

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