Expression Parser Test

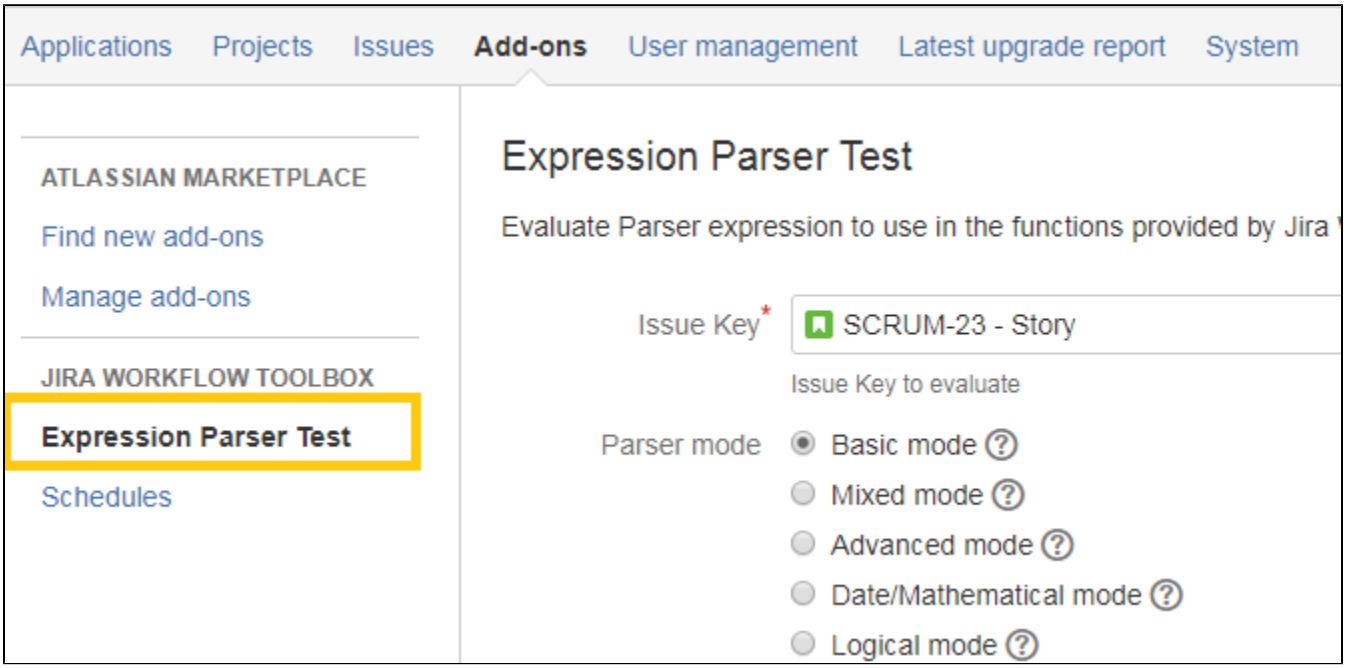
On this page

- Expression Parser Test
- Usage
- Example
- Related Features

SINCE VERSION 2.4.0

Expression Parser Test

The Expression Parser Test is located in the Jira Administration under **Add-ons Jira Workflow Toolbox Expression Parser Test**.



Usage

The Expression Parser Test page provides the capability to test/evaluate JWT expressions against existing issues. Besides the issue key to evaluate, you can choose one of the five different parser modes described below.

⚠️ Caution: the Expression Parser test page does **not** support all the JWT field codes normally found in workflow functions. Field codes for transitional fields e.g. *Transition comment*, are not supported.

Issue Key

All issue related expressions will be evaluated using the chosen issue.

Parser Mode

The chosen parser mode determines the parser functionality and any data returned by the evaluation.

Mode	Function	Result
Basic Mode	Field codes will be returned with any relevant values found in the chosen issue. Basic mode does **not** support complex parser expressions.	String
Mixed Mode	Supports string and math-time expressions in so far these are coded with 3 curly brackets, i.e. {{{expression}}}.	String

Advanced Mode	Supports all Field codes and Expression Parser functions.	String
Date/Mathematical mode	Calculates date/time or numerical values.	Number
Logical mode	Returns the result of a Boolean expression.	Boolean
Issue List mode	Used for **selecting issues** in many features like post-functions Transition issues and Update fields.	Issue List

Expression

Syntax to be evaluated/interpreted by the Expression Parser.

Check Syntax

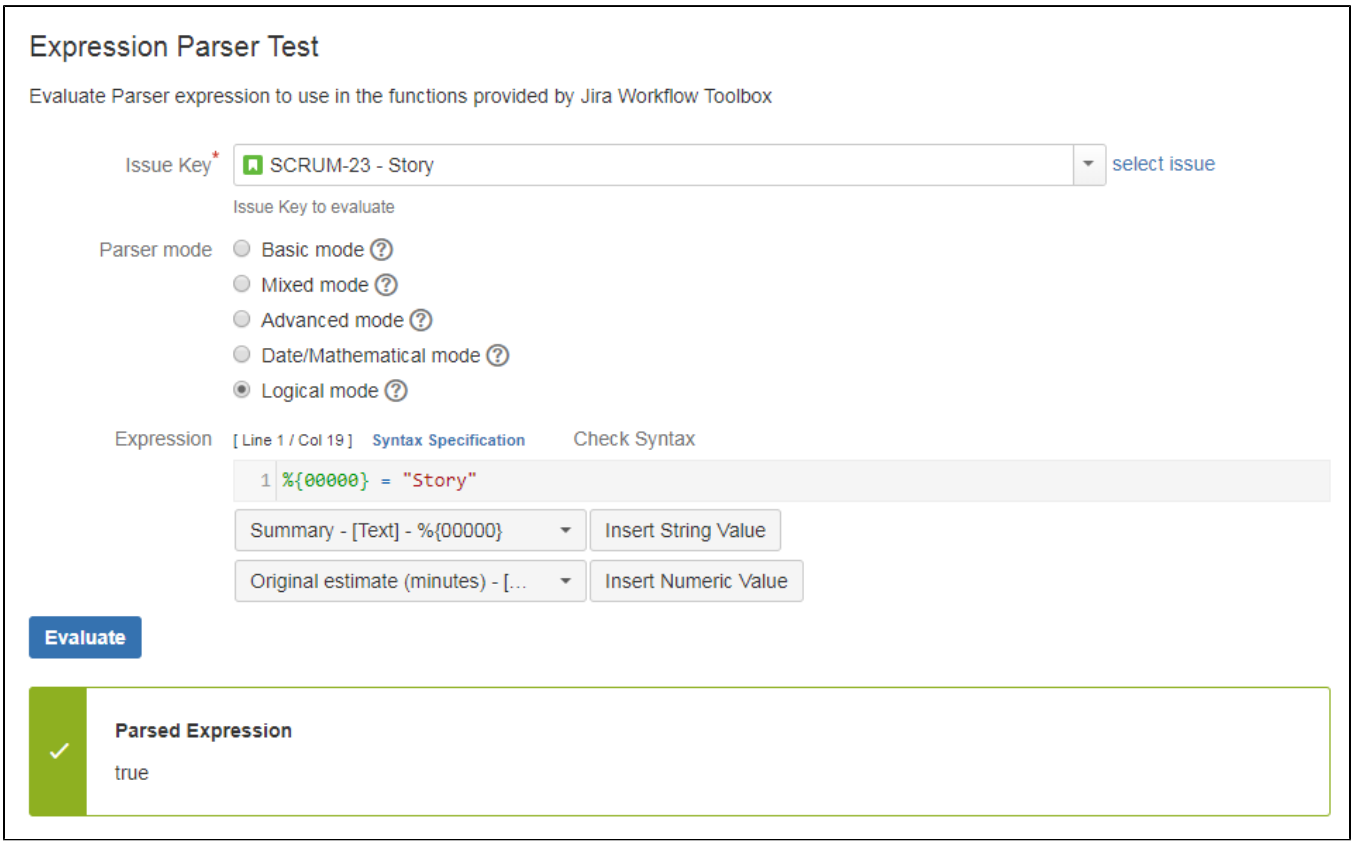
The JWT Syntax check is supported for the following 3 modes: Advanced Mode, Date/Mathematical mode, Logical mode. Syntactical errors will be displayed with a corresponding error message.

Example

An issue with issue key SCRUM-23 was chosen. The expression to be evaluated in *Logical Mode* is whether or not the issue field Summary contains the value "*Story*".

> *%{00000} = "Story"*

In this case the parser will return ***true*** (where %{00000} is the field code for Summary) as shown below.



Related Features

- Expression Parser
- Field Codes