

# JWT Release Notes 2.5.0

## Direct download from Marketplace

### Automation Toolbox for Jira

This **major release** ships with a new **library** that comes bundled with Jira Workflow Toolbox. The **xApps Library** is used to commonly provide **expression parser features** for JWT and several upcoming apps.

The first new member of the **toolbox family** using the library is **Automation Toolbox for Jira**, which unchains the power of the parser from its current workflow scope.

Check out the [rich set of functions](#) and start automating your daily tasks in Jira today!

## New features

- Move issues on a transition using the [Move issues](#) post function
- Use advanced parsing mode for composing meaningful error messages in all validators
- Added the xApps Library to use the Jira Workflow Toolbox Expression Parser in several upcoming apps

## New expressions in the parser

Function	Returned value
<b>allCommentDates()</b> : number list Available since version 2.5.0	Returns the dates of related comments as a number list.
<b>allCommentDates(string issue_keys)</b> : number list Available since version 2.5.0	Returns the dates of related comments from the entered <b>issue_keys</b> as a number list.  Example: <b>allCommentDates(["SW-1", "SW-2"])</b> returns a list with all the comment dates for issues with the key <b>SW-1</b> and <b>SW-2</b> .
<b>allCommentDates(issue list issue_list)</b> : number list Available since version 2.5.0	Returns the dates of related comments in the entered <b>issue_list</b> as a number list.  Example: <b>allCommentDates(issuesFromJQL("project = softwareProject"))</b> returns a list with all the comment dates for all issues in project <b>softwareProject</b> .
<b>projectLeader(string projectKey)</b> : string Available since version 2.5.0	Returns the project lead of the <b>projectKey</b> .  Example: <b>projectLeader("SW")</b> returns the project lead key from the project with the key <b>SW</b> .
<b>status(number id)</b> : string Available since version 2.5.0	Returns the name of the status with the id <b>id</b> .  Example: <b>status(1)</b> returns the status name with the id <b>1</b> .
<b>resolution(number id)</b> : string Available since version 2.5.0	Returns the name of the resolution with the id <b>id</b> .  Example: <b>resolution(10000)</b> returns the resolution name with the id <b>10000</b> .
<b>issueType(number id)</b> : string Available since version 2.5.0	Returns the name of the issue type with the id <b>id</b> .  Example: <b>issueType(10000)</b> returns the issue type name with the id <b>10000</b> .
<b>option(number id)</b> : string Available since version 2.5.0	Returns the name of the option with the id <b>id</b> .  Example: <b>option(10000)</b> returns the option name with the id <b>10000</b> .
<b>priority(number id)</b> : string Available since version 2.5.0	Returns the name of the priority with the id <b>id</b> .  Example: <b>priority(1)</b> returns the priority name with the id <b>1</b> .
<b>project(number id)</b> : string Available since version 2.5.0	Returns the key of the project with the id <b>id</b> .  Example: <b>project(10000)</b> returns the project key of the project with the id <b>10000</b> .
<b>issueSecurityLevel(number id)</b> : string Available since version 2.5.0	Returns the name of the issue security level with the id <b>id</b> .  Example: <b>issueSecurityLevel(10000)</b> returns the issue security level name with the id <b>10000</b> .

<b>invertList</b> (number list <b>I</b> ) : number list  Available since version 2.5.0	Returns <b>I</b> in inverted order.  Example: <code>invertList([1, 2, 3])</code> returns number list <b>[3, 2, 1]</b> .
<b>invertList</b> (string list <b>I</b> ) : string list  Available since version 2.5.0	Returns <b>I</b> in inverted order.  Example: <code>invertList(["first", "second", "third"])</code> returns string list <b>["third", "second", "first"]</b> .

## Improvements

- Reference standard objects like issue types, statuses or resolutions by ID, e.g. use `status(1)` instead of `"Open"`
- Use IDs for custom field options, e.g. use `option(10001)` instead of `"Option A"`
- Skip validations for issues being created by email
- Set the assignee to **Automatic** when updating fields
- Mark all values in a multi-valued field as required in the [Fields required or changed validator](#) validator
- Use **Standard** option for fields in the [Create issues and sub-tasks](#) post function
- Parse text for selecting transitions and statuses within the [Transition issues](#) post function

## Bug fixes

- When using "IN" operator, **options containing commas** didn't work
- The **field code injector** in the [Add a comment](#) post function only displayed user fields
- The [Copy parsed text to a field](#) post function added a user **twice as request participant** in some circumstances
- Fixed workflow exception in the [Create issues and sub-tasks](#) post function when creating a new issue from a **SD issue** which contains the **Satisfaction** field
- Fixed **UI behavior** when editing **large schedules**
- Fixed error when trying to set the **Sprint ID** in the [Set a field as a function of other fields](#) post function
- **Copying multi-select fields** when using the [Create issues and sub-tasks](#) post function didn't work properly
- Fixed parser expression check **via REST call** for large expressions
- Field code for available **target statuses** threw a `NullPointerException`
- Using function `fieldValue()` with Insight Objects custom fields didn't work
- Selecting a standard option caused the **post function view** to break and made the configuration unavailable in some circumstances
- Not all **math expressions** used to work in the [Send an email](#) post function
- Setting the epic link on [Create issues and sub-tasks](#) didn't work
- Fixed **wrong popup** window when saving a schedule in **Firefox** and **Internet Explorer**
- Fixed **field code injector** for several custom field codes
- Copying multi-select fields in the **Create Issue post function** didn't work properly
- **Syntax highlighting** didn't work when Insight was enabled
- Fixed parser check error when using the **Sprint** field in a calculated field
- **Customer request channel** couldn't be resolved in **Jira 8**