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

invertList(number list **l**) : number list Available since version 2.5.0	Returns **l** in inverted order. Example: **`invertList([1, 2, 3])`** returns number list **[3, 2, 1]**.
invertList(string list **l**) : string list Available since version 2.5.0	Returns **l** in inverted order. Example: **`invertList(["first", "second", "third"])`** returns string list **["third", "second", "first"]**.

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 injecto**r 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 mutli-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**