Update fields

The **Update fields** post function automatically updates a **single** or **multiple** issue fields on one or multiple target issues.

Additional **field values** can be **inherited** or manually set using the JWT expression editor.

Configuration

Target issue

The issues to be updated. If a list of issue keys is passed, the update will be performed for every issue in the list.
The following options are available:

- Current issue
- Parent issue
- Parser expression

When the option **parser expression** is selected, an expression parser input field is displayed and the input has to be an **issue key** or **an issue list**. Every issue in the given list will be updated.

Fields

Every supported field can either be set or cleared.

After selecting a field and clicking the **Add** button, you can select in the **popup** how to set the value. There are three or - when dealing with selectable fields - four options:

- Copy field from current issue
- Clear field value
- Set field value manually - Read more about the JWT expression editor
- The option **Selected value** is available for the following fields types:
    - Jira Software related fields like **Sprint** or **Epic**
    - User related fields like **Assignee** or **Reporter**
    - Version related fields like **Affects-** or **Fix version/s**
    - Component/s
    - Labels
    - Priority
    - Resolution
    - Security level
    - and all option based custom fields that are supported

Run as

Select the user that will be used to execute the post function. By default, it is set to the current user that executes the transition.

The following options are available:

Option	Description
Selected user	Select a specific Jira user.
User in field	Select the field containing the user that will be used to execute the post function.

The configured user must have all necessary permissions to transition the target issue.

Conditional execution

You can **optionally** specify a logical expression or a Jira expression depending on the chosen Parsing mode to define the circumstances (or conditions) under which the post function should be executed.

The result of the logical expression must return a boolean value of either:

- `true` the post function will be executed
- `false` the post function will **not** be executed

Using the **conditional operator,** even complex or multi-layered conditions can be constructed.

Make sure to learn more about defining logical expressions and browse through the various **examples** here: Logical mode

Use cases and examples

Use case	JWT feature	Workflow function	Parser functions	Use case description	Complexity
Add a sub-task's summary and key to the description of its parent		Update fields		When a **sub-task is created**, its **summary**, **issue key** and **date, and time of creation** will be added to the **description** of the **parent** issue.	BEGINNER
Add three days skipping weekends automatically to a Date Picker		Update fields	dateTimeToString() dayOfTheWeek() addDays()	Add three days to a *Date Picker* field from the current date.	BEGINNER
Add watchers from another field		Update fields Transition issue	union() toStringList()		
Assign an issue to the project lead, if the issue is unassigned on creation		Update fields		When an issue is **created without an assignee** selected, the issue will be assigned to the **project lead** of the project.	BEGINNER
Assign an issue to the user who last commented on it		Update fields		Assign the issue to the **user** who **last commented** on the issue.	BEGINNER
Assign important issues to the project lead		Update fields		Automatically **assign** and issue to the **project lead**. Issues will **only** be re-assigned if the **priority** of the issue is set to **Highest** to make sure that only important issues are being escalated.	BEGINNER
Copy field values from epic to issues under it after creation		Update fields Transition issue	epic() issuesUnderEpic() filterByStatus()	**Copy the values** of the **fields** defined in the **Update fields** post-function **to the issues under the epic** after their **creation**.	BEGINNER
Copy labels of a sub-task to the parent issue upon closing		Update fields		When a sub-task is closed, the **labels** of the **sub-task** will be added to the **Labels** field of the **parent issue**.	BEGINNER

Keep parent's priority in sync		Update fields		Set the **priority** of the **parent issue** to the **priority** of the **current issue**.	BEGINNER
Obtain the difference between two dates		Update fields		Update a number field with the difference in days between two dates obtained from two *Date Picker* fields.	BEGINNER
Set a date field to a future date		Update fields	dateTime ToString()	Set any date field of an issue to a future date.	BEGINNER
Set a date field to the current date		Update fields	dateTime ToString()	Set a Date Time Picker field to the current date	BEGINNER
Set Fix version to Affects version when resolving an issue		Update fields		When an **issue is resolved** and the resolution set to "**Done"** the **Affects version/s** will be added to the **Fix version/s** field.	INTERMEDIATE
Set the priority to Highest if the 'Infrastructure' component is selected		Update fields		When an issue is **created** with the "**Infrastructure" component** selected, the priority will be set to "**Highest**".	BEGINNER

If you still have questions, feel free to refer to our support team.