

Create sub-tasks depending on selected values in custom field

On this page

[Configuration](#) | [Screenshots](#) | [Related use cases](#)

Use case

Creating **repetitive issues** might be **time-consuming work**, especially when done on a daily basis. This post function can be used for **pre-configuring a set sub-tasks**.

Each time the transition is executed, all needed sub-tasks will be created **depending on the selected values in a checkbox field**.

Should I use a workflow post function or an automation rule?

When to use this workflow post function and when to use an automation rule?

This use case is very useful, if repetitive issues have to be created in only one project or workflow. Then a post function is the easier way to go.

But if this setup is needed for several projects, the [Create issue action](#) would make more sense. In this case you might want to check out our [co responding automation use case](#).



Configuration

1

Add a [Create issue](#) post function to the desired **workflow transition**.

The **Create issue transition** might be here the best choice. When added to the **Create issue transition**, the post function will be automatically executed each time an issue is created.

2

Issues to be created

1

Mode

Choose **Multiple issues**

2

Expression

Enter the following parser expression

```
toStringList(%{issue.cfnnnnn})  
# Replace the field code with the one of your custom field. Use the field  
dropdown to automatically insert the field code.  
# You can use any field supporting multiple options, e.g. checkboxes or multi-  
selects.
```

with parsing mode set to **Text list**

3

Issue type

Choose **Selected issue type** **Sub-task**

4

Parent issue

Choose **Current issue**

3

Summary

Enter the following expression

```
%{seed.text}  
# This is a reference to each value returned by the text list.
```

with parsing mode set to **Advanced Text**

4

Description

To enrich the issue with further information, you might want to add a short description. It is not mandatory, though.

5

Run as

User

Choose **Current user**



Screenshots

Update parameters of the Create issue Function for this transition.

Update parameters of the Create issue Function for this transition.

Basic configuration *

Define the required parameters needed to execute this post function.

Mode *
Multiple issues
Select the number of issues to be created.

Expression *
1 toStringList(issue.labels)
The expression must return a text list. The number of list elements defines the number of issues to be created. From here on, seed texts can be referenced by %seed.text).

Issue type *
Selected issue type
Select the issue type for the issue to be created.
Sub-task

Parent issue *
Current issue
Select the parent of the issue to be created.

Fields *

Define the fields to be set.

Summary *
1 %
Enter plain text and optionally use field codes, e.g. %issue.summary, to insert field values.

Description
1
Enter plain text and optionally use field codes, e.g. %issue.summary, to insert field values.

Additional fields

Optionally define further fields to be set.

Field	Value
No fields configured!	
Add field	

Copy remaining fields
Don't copy fields
Select whether to copy remaining fields from a specific issue.

- Issue links
- Additional options
- Conditional execution
- Run as

Related use cases

Use case	JWT feature	Workflow function	Parser functions	Label
Clone epic, tasks and sub-tasks		Create issue	toString() toStringList() issueKeysToIssueList() replaceFirst() first() fieldValue() textOnStringList() findModify() subtasks() issuesUnderEpic()	

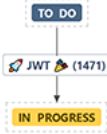
Create issue

Documentation

Create one or multiple issues or sub-tasks.

Workflow

JWT 3.0.0 Use cases



Examples

Pick one of the following example configurations and get started in no time. More details will be provided upon your selection.

- Create an issue in the current project
- Create a sub-task for each component
- Create a sub-task for high priority issues

Create an issue in the current project		Create issue		STAFF PICK
Create an issue with a custom summary		Create issue	previousValue()	
Create a sub-task for each component		Create issue	toStringList()	
Create a sub-task for every recurring deadline within a task		Create issue	addMonths() getMatchingValue() dateToString()	
Create a sub-task for every sub-task closed		Create issue	findReplaceFirst() first() findPattern() toString() sum() toNumber()	
Create a sub-task for every user in a field and assign them		Create issue	getMatchingValue() toStringList()	
Create child issue within Advanced Roadmaps hierarchy		Create issue		
Create issues randomly		Create issue	modulus() round() random()	
Create issue under epic		Create issue	toString() epic()	
Create several issues combining fields		Create issue	nthElement() toStringList() modulus() count() ceil()	
Create several sub-tasks depending on the component		Create issue		
Create sub-tasks depending on selected values in custom field		Create issue	toStringList()	STAFF PICK
Create three issues with individual summaries		Create issue	getMatchingValue()	

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