Add sub-tasks to an automatically created issue

On this page

Use case

Creating multiple issues can be time consuming and on top of it, if you need to add sub-tasks to those newly created issues could even lead to mistakes, what is expected from redundant work!

In this use case. we will save your time by using our Create issue post function in a collaboration with J WT automation rules. The parent issue will be **created by the post function** when a status transition has been made, this will **trigger the automation rule** to create the sub-tasks.

Create issue post function

1. Add the create issue post function to the desired workflow transition.
2. Issues to be created
   1. **Mode**

      Choose **Single issue**
   2. **Issue type**

      Choose **Selected issue type Task**
   3. **Project**

      Select the project you want to create the issue in from the given options. (**Current project** by default)
3. **Summary**

   Enter a summary that fits your needs, e.g.

   ```
   Internal progress report
   ```

4

Description

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

5

Further configuration

Fields

You have the possibility to set additional fields on the issue to be created.

Maybe you want to predefine the issue **Assignee** or a **time estimation** for time logging purposes.

Don't want to set the fields? You can also copy existing values from the current issue!

For further information, you might want to take a look at the **Create issue documentation.**

Issue links

This is an optional parameter where you can define issue links for the issue to be created.

Additional options

If you want to copy issue keys of created issues into the field **Temporaty Text 3** as a text list, choose this option.

Conditional execution

If you want some conditions that need to be checked before the execution of this post function, add your logical expression.

Uncertain about this part? **Check out our documentation!**

Run as

Select which **user** will be used to execute this post function. By default this parameter is set to the **current user**. You can also use field codes to run the function as a dynamic user (e.g. current assignee).

Make sure that the user running the post function has all the **relevant permissions** to perform the actions defined in the configuration (e.g. "Update Issues")!

If you want to keep track the actions being performed automatically, we suggest to create a **dedicated JWT account**, granted all relevant **permissions**, and use it in the Run as parameter to identify which changes have been made with JWT.

Automation rule

Issue Created event

1

Create a new rule and name it appropriately.

> Providing a description will help you to identify what the rule does but this step is **optional**.

2

Add a **Trigger Issue Event Issue Created**

Boolean condition

3

Add the **Boolean condition** and enter the following **Parser Expression***

```
%{trigger.issue.project.key} = "key"
```

> Replace **key** with your **project key**

> The Boolean condition can be updated to meet your conditions on the sub-tasks creation.

Create issue action

4

Issue Details

1

Issues to be created

Choose **Multiple issues**

2

Parser Expression

```
3
```

with parsing mode set to **Numeric**

This leads to creating 3 sub-tasks, you can change the number to create as many sub-tasks as you need.

3

Issue type

Choose **Sub-task**

4

Parent issue

Choose **Trigger issue**

Fields

1

Summary

Use the following **Parser Expression**:

```
getMatchingValue(^,[1,2,3],
["Documentation","Marketplace","Report"])
```

with parsing mode set to **Advanced text**.

This will assign the summaries of the sub-tasks in the respective order as in the expression. e.g. **Sub-task 1** would have the summary as **Documentation**.

2

Description

Use the following **Parser Expression**:

```
getMatchingValue(^,[1,2,3],
["Description 1","Description 2","Description 3"])
```

with parsing mode set to **Advanced text**.

5

Enable the rule by clicking on the **Enable button** 

Import the example

Import the **JSON** file below to get started in no time.

JSON

After importing the JSON file, make sure to **check** the **configuration** of the rule. Non-existing configuration elements (issue types, fields, values etc.) will be highlighted.


```
{
"name": "Add sub-tasks on issue creation",
"description": "",
"creator": "admin",
"status": true,
"triggerData": "1",
"triggerType": "ISSUE_EVENT",
"configuration": {
"refs": [
"issue",
"system",
"trigger.issue",
"trigger.parent"
],
"triggerType": "1"
},
"children": [
{
"sequence": 0,
"type": "BOOLEAN_CONDITION",
"ruleEntityType": "CONDITION",
"configuration": {
"refs": [
"issue",
"project",
"system",
"trigger",
"trigger.issue",
"trigger.parent"
],
"expression": "%{trigger.issue.project.key} = \"key\"",
"expressionParsingMode": "logical",
"actingUser": "field_00020"
},
"children": [
{
"sequence": 0,
"type": "CREATE_ISSUE",
"ruleEntityType": "ACTION",
"configuration": {
"refs": [
"issue",
"project",
"system",
"trigger",
"trigger.issue",
"trigger.parent",
"seed.number"
],
"issueSelectionParserValue": "3",
"issueSelectionParserValueParsingMode": "math",
"issuetype": "10001",
"parentSelection": "eventIssue",
"parentIssueKeyParsingMode": "textBasic",
"projectKeyParsingMode": "textBasic",
"actingUser": "field_00020",
"summary": "getMatchingValue(^,[1,2,3],\r\n[\"Documentation\",\"Marketplace\",\"Report\"])\r\n\r\n",
"summaryParsingMode": "textAdvanced",
"description": "getMatchingValue(^,[1,2,3],\r\n[\"Description 1\",\"Description 2\",\"Description 3\"])",
"descriptionParsingMode": "textAdvanced",
"inheritFieldsFrom": "none",
"inheritFieldsIssueKeyParsingMode": "textBasic",
```

```
"inheritIssueLinks": "none",
"inheritIssueLinksIssueKeyParsingMode": "textBasic",
"fields": [],
"issueLinks": [],
"issueSelection": "math",
"inheritFields": []
},
"children": null,
"hasChildren": false
}
],
"hasChildren": true
}
],
"hasChildren": true
}
```


Related use cases

Title	Automated action	JWT feature	Label
Add sub-tasks to an automatically created issue	Create issue action		
Add sub-tasks to an issue on creation	Create issue action		
Create sub-tasks depending on selected values in a custom field	Create issue action		
Manually create test issues	Create issue action		

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