

# Add sub-tasks to an issue on creation

## On this page

[Issue Created event](#) | [Boolean condition](#) | [Create issue action](#) | [Screencast](#) | [Import the example](#) | [Related use cases](#)

## Use case

Often creating **repetitive issues** is daily work, e.g. in a software development release process. It might be that for every new feature specific tasks have to be done in order to successfully release a new software version containing this feature, be it updating the product documentation, the marketplace presence or sending a newsletter to all customers.

To **avoid creating these issues always manually**, the following automation rule can easily manage this by **creating the issues automatically**.

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

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

This use case is very useful, if repetitive issues have to be created in several projects that have different workflows. Configuring just one automation rule might then be the easier way to go.

But if this scenario is only necessary for one project and workflow, the [Create issue post function](#) would make more sense. In this case you might want to check out our [corresponding post function use case](#).



## 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} in ["key1", "key2", "key3"] and %{trigger.issue.issueType} = "New Feature"
```

Replace **key1**, **key2** and **key3** with your **project keys**.



## Create issue action

4

### Issue Details

1

#### Issues to be created

Choose **Multiple issues**

2

#### Parser Expression

3

with parsing mode set to **Date/Mathematical**

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","Newsletter"])
```

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

2


#### Description

Use the following **Parser Expression**:

```
getMatchingValue(^,[1,2,3],
["Issue for creating the documentation for the feature",
"Issue for updating the marketplace presence for the app",
"Issue for creating a newsletter for the feature"])
```

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

5

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



## Screencast

This is how the configuration above should look on your screen

Your browser does not support the HTML5 video element



## 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 to issue on creation",
  "description": "",
  "creator": "admin",
  "status": false,
  "triggerData": "1",
  "triggerType": "ISSUE_EVENT",
  "configuration": {
    "refs": [
      "issue",
      "system",
      "trigger.issue",
      "trigger.parent"
    ],
    "triggerType": "1"
  },
  "children": [
    {
      "sequence": 0,
      "type": "CREATE_ISSUE",
      "ruleEntityType": "ACTION",
      "configuration": {
        "refs": [
          "issue",
          "project",
          "system",
```






```

        "trigger",
        "trigger.issue",
        "trigger.parent",
        "remote.math"
    ],
    "issueSelectionParserValue": "3",
    "issueSelectionParserValueParsingMode": "math",
    "issuetype": "5",
    "parentSelection": "eventIssue",
    "parentIssueKeyParsingMode": "textBasic",
    "projectKeyParsingMode": "textBasic",
    "actingUser": "field_00020",
    "summary": "getMatchingValue(^,[1,2,3],\r\n[\"Documentation\", \"Marketplace\", \"
Newsletter\"])",
    "summaryParsingMode": "textAdvanced",
    "description": "getMatchingValue(^,[1,2,3],\r\n[\"Issue for creating the documentation for
the feature\", \"Issue for updating the marketplace presence for the app\", \"Issue for creating a
newsletter for the feature\"])",
    "descriptionParsingMode": "textAdvanced",
    "inheritFieldsFrom": "none",
    "inheritFieldsIssueKeyParsingMode": "textBasic",
    "inheritIssueLinks": "none",
    "inheritIssueLinksIssueKeyParsingMode": "textBasic",
    "fields": [
        {
            "field": {
                "label": "Assignee",
                "value": "00003",
                "meta": {
                    "isStringEphemeral": false,
                    "isCustomField": false,
                    "isIndexable": false,
                    "isCalculated": false,
                    "isMultiValued": false,
                    "type": "User",
                    "isNumericEphemeral": false,
                    "isSystemField": true,
                    "hasStandardConfig": true,
                    "isSubtaskOnlyField": false,
                    "isMultiWritable": false,
                    "isNumericField": false,
                    "isVirtualField": true,
                    "id": "3",
                    "key": "assignee",
                    "isClearable": true
                }
            },
            "valueType": {
                "label": "Value returned by parsed expression",
                "value": "parser"
            },
            "value": {
                "label": "% Creator (Trigger)",
                "value": "%{trigger.issue.creator}"
            },
            "parsingMode": {
                "label": "Advanced",
                "value": "textAdvanced"
            }
        }
    ],
    "issueLinks": [],
    "issueSelection": "math",
    "inheritFields": []
},
"children": null,
"hasChildren": false
}
],
"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.