

# Manually create test issues

On this page

[Manual trigger](#) | [JQL Selector](#) | [Create issue action](#) | [Screencast](#) | [Import the example](#) | [Related use cases](#)

## Use case

Sometimes it might be helpful to have customized test issues at hand for several scenarios; testing new feature implementations, new project setups or simply new JWT automation rules.

The following rule will take care of **Creating customized issues** for testing anything that needs to be tested.



## Manual trigger


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 Manual**

No further configuration needed. The rule will only be executed if a user hits the Execute button .



## JQL Selector

3

Add **Selector JQL Selector**

4

## JQL Query

Use the following **Expression**:

```
key = nnnnn
```

**nnnnn** has to be an issue key. In this case we are going to create test issues as **sub-tasks** under a **predefined issue**.

But you can also adapt the configuration to create tasks in a specific project/ under an epic/ etc.



## Create issue action

5

Next to the **JQL Selector** click on **Add Action Create issue**

### Issue Details

1

#### Issues to be created

Choose **Multiple issues**

2

#### Parser Expression

Enter the following expression with mode set to **date/ mathematical**:

3

3

#### Issue type

Choose one of the available sub-task issue types

4

#### Issue type

Choose **Selected issue**

### Fields

1

#### Summary

Enter the following Expression with mode set to **Advanced text**:

++++TESTING+++ Sub-task " + ^

2

#### Description

Enter the following Expression with mode set to **Basic text**:

Sub-task for testing purposes

#### Fields



Set the fields according to your needs.



## 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": "# Create test issues",
  "description": "",
  "creator": "admin",
  "status": false,
  "triggerData": "",
  "triggerType": "MANUAL",
  "configuration": {
    "refs": [
      "issue",
      "system",
      "trigger.issue",
      "trigger.parent"
    ],
    "actingUser": "field_00020",
    "triggerType": ""
  },
  "children": [
    {
      "sequence": 0,
      "type": "JQL_SELECTOR",
      "ruleEntityType": "SELECTOR",
      "configuration": {
        "refs": [
          "system"
        ],
        "jql": "key = EO-57",
        "jqlParsingMode": "jql",
        "actingUser": "field_00020"
      },
      "children": [
        {
          "sequence": 0,
          "type": "CREATE_ISSUE",
          "ruleEntityType": "ACTION",
          "configuration": {
            "refs": [
              "issue",
```

```






        "issues",
        "selector.issue",
        "selector.parent",
        "system",
        "remote.math"
    ],
    "issueSelectionParserValue": "3",
    "issueSelectionParserValueParsingMode": "math",
    "issuetype": "5",
    "parentSelection": "selectorIssue",
    "parentIssueKeyParsingMode": "textBasic",
    "projectKeyParsingMode": "textBasic",
    "actingUser": "field_00020",
    "summary": "\\+++ TESTING +++ Sub-task \\\" + ^",
    "summaryParsingMode": "textAdvanced",
    "description": "Sub-task for testing purposes",
    "descriptionParsingMode": "textBasic",
    "inheritFieldsFrom": "none",
    "inheritFieldsIssueKeyParsingMode": "textBasic",
    "inheritIssueLinks": "none",
    "inheritIssueLinksIssueKeyParsingMode": "textBasic",
    "fields": [
        {
            "field": {
                "label": "Priority",
                "value": "00017",
                "meta": {
                    "isStringEphemeral": false,
                    "isCustomField": false,
                    "isIndexable": false,
                    "isCalculated": false,
                    "isMultiValued": false,
                    "type": "Issue priority",
                    "isNumericEphemeral": false,
                    "isSystemField": true,
                    "hasStandardConfig": true,
                    "isSubtaskOnlyField": false,
                    "isMultiWritable": false,
                    "isNumericField": true,
                    "isVirtualField": true,
                    "id": "17",
                    "key": "priority",
                    "isClearable": true
                }
            },
            "valueType": {
                "label": "Selected value",
                "value": "fix"
            },
            "value": {
                "label": "High",
                "value": "10001"
            },
            "parsingMode": {
                "label": "",
                "value": ""
            }
        }
    ],
    "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	 	
Create sub-tasks depending on selected values in a custom field	Create issue action		
Add sub-tasks to an issue on creation	Create issue action		
Manually create test issues	Create issue action		

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