

Close stories when epic is done

On this page

[Issue transitioned event](#) | [Boolean condition](#) | [Issue selector](#) | [Boolean condition](#) | [Transition issue](#) | [Screencast](#) | [Import the example](#) | [Related examples](#)

Use case

Avoid all the manual work by **automatically closing stories when the epic is completed**.



Issue transitioned 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 transitioned event**



Boolean condition

3

Add the following boolean expression:

```
%{trigger.issue.status} = "Done"
```



Issue selector

4

Next to the **JQL condition** click on **Add Selector** **Issue selector**.

1

Target issue(s)

Select **Issues under epic/ stories**



Boolean condition

5

Next to the **Issue selector** click on **Add Condition Boolean condition**

1

Add the following expression:

```
%{selector.issue.status} != "Done"
```



Transition issue

6

Next to the **JQL condition** click on **Add Action Transition issue**

1

Mode


Choose **transition to status**

2

Status

Choose **Done**

7

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



Screencast

No screencast available at the moment.



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": "Close stories when epic is done",
  "description": "",
  "creator": "admin",
  "status": false,
  "triggerData": "",
  "triggerType": "ISSUE_TRANSITIONED_EVENT",
```

```

"configuration": {
  "refs": [
    "issue",
    "system",
    "trigger.issue",
    "trigger.parent"
  ],
  "triggerType": ""
},
"children": [
  {
    "sequence": 0,
    "type": "BOOLEAN_CONDITION",
    "ruleEntityType": "CONDITION",
    "configuration": {
      "refs": [
        "issue",
        "project",
        "system",
        "trigger",
        "trigger.issue",
        "trigger.parent"
      ],
      "expression": "%{trigger.issue.status} = \"Done\"",
      "expressionParsingMode": "logical",
      "actingUser": "field_00020"
    },
    "children": [
      {
        "sequence": 1,
        "type": "ISSUE_SELECTOR",
        "ruleEntityType": "SELECTOR",
        "configuration": {
          "refs": [
            "issue",
            "project",
            "system",
            "trigger",
            "trigger.issue",
            "trigger.parent"
          ],
          "option": "stories",
          "issueListExpressionParsingMode": "issues",
          "actingUser": "field_00020"
        },
        "children": [
          {
            "sequence": 1,
            "type": "BOOLEAN_CONDITION",
            "ruleEntityType": "CONDITION",
            "configuration": {
              "refs": [
                "issue",
                "issues",
                "project",
                "selector.issue",
                "selector.parent",
                "system",
                "trigger",
                "trigger.issue",
                "trigger.parent"
              ],
              "expression": "%{selector.issue.status} != \"Done\"",
              "expressionParsingMode": "logical",
              "actingUser": "field_00020"
            },
            "children": [
              {
                "sequence": 1,
                "type": "TRANSITION_ISSUE",
                "ruleEntityType": "ACTION",

```






```

        "configuration": {
          "refs": [
            "issue",
            "issues",
            "project",
            "selector.issue",
            "selector.parent",
            "system",
            "trigger",
            "trigger.issue",
            "trigger.parent"
          ],
          "option": "status",
          "status": "10012",
          "actingUser": "field_00020"
        },
        "children": null,
        "hasChildren": false
      },
    ],
    "hasChildren": true
  },
  ],
  "hasChildren": true
},
],
"hasChildren": true
},
],
"hasChildren": true
}

```



Related examples

Title	Automated action	JWT feature	Label
Assign an issue to the last assignee on a transition	Assign issue action		
Assign issues to your team during a transition	Assign issue action		
Automatically assign issues to your team	Assign issue action		
Re-assign issues and leave a comment	Add comment action Assign issue action		
Update assignee and reporter to facilitate user deletion	Add comment action Assign issue action Update field action		STAFF PICK

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