

Assign new issues to a different project role depending on field value in current issue

On this page

- [Features used to implement the example](#)
- [Example: Assign new issues to a different project role depending on field value in current issue](#)
- [Other examples of that functions](#)
- [Related Usage Examples](#)

Features used to implement the example

- [Create issues and sub-tasks](#)

Example: Assign new issues to a different project role depending on field value in current issue

First of all, using groups in a workflow impairs its reusability. It's much better to use **project roles**. You should use groups only for assigning users to project roles. As you surely know, Jira issues can only be assigned to **one single user** or to be **unassigned**.

[Jira Workflow Toolbox](#) implements a functionality for [setting a default user for a project role per project](#). Once you have done this using [project properties](#) or [user properties](#), you can assign your issues or sub-tasks created by [Create issues and subtasks](#) post-function as I explain in the following example.

I explain you how to assign issues according to **Priority** of **current issue** (the issue that is executing the post-function). The behavior implemented in the example is:

If **Priority** or current user is **Critical** or **Blocker** assign new issue to project role **Managers**.

If **Priority** or current user is **Major** assign new issue to project role **Administrators**.

Otherwise assign new issue to project role **Developers**.

Target field to be set in the new issue:	Assignee
<div><div>Value type:</div><div>Parsed text (advanced mode) ▾</div><div>Way used to define the value used for setting the target field.</div></div>	
<div><div>Text to be parsed:</div><div><div>Advanced Parsing Mode (Syntax Specification and Examples) [Line 1 / Col 120]</div><div>1 <code>getMatchingValue(#{00017}, ["Blocker", "Critical", "Major", ["Managers", "Managers", "Administrators", "Developers"])</code></div><div><div>Advanced parsing mode: Strings literals are written in double quotes, e.g., "This is a string.". Operator '+' is used to concatenate strings, and field codes are like in basic mode, e.g., "Issue key is " + #{00015} + ".". More information at parser syntax documentation.</div><div><div>String Field Code Injector:</div><div>Priority - [Issue priority] - #{00017} ▾</div><div>Numeric/Date Field Code Injector:</div><div>Original estimate (minutes) - [Number] - {00068} ▾</div><div>Field code injectors reference: <input checked="" type="radio"/> Current issue <input type="radio"/> Seed issue <input type="radio"/> Parent of new sub-task</div></div></div></div></div>	

Once configured, you have:

Set Fields:
Sets field values in the new issues.

Field to be set:

Reporter - [User]

Add

Field	Type of Value	Value	Actions
Assignee	Parsed text (advanced mode)	getMatchingValue(%{Priority}, ["Blocker", "Critical", "Major"], ["Managers", "Managers", "Administrators", "Developers"])	<div>Edit</div> <div>Remove</div>

String expression is:

```
getMatchingValue(%
{00017}, ["Blocker", "Critical", "Major"], ["Managers", "Managers", "Administrators", "Developers"])
```

Note that:

- **%{00017}** is field code for **Priority**

Other examples of that functions

Create issues and sub-tasks

Page: [Assign new issues to a different project role depending on field value in current issue](#)
Page: [Clone an issue and all its subtasks \(with additional restrictions\)](#)
Page: [Create 3 issues in 3 different projects](#)
Page: [Create a dynamic set of sub-tasks based on checkbox selection with unique summaries](#)
Page: [Create a static set of sub-tasks with unique summaries](#)
Page: [Create a story for each component in an epic](#)
Page: [Create a sub-task for each user selected in a Multi-User Picker](#)
Page: [Create a sub-task in each story of an epic](#)
Page: [Create specific sub-tasks for each selected component](#)

Related Usage Examples

- [Creating a Jira Service Desk internal comment](#)
 - [example](#)
 - [post-function](#)
- [Limit the number of hours a user can log per day](#)
 - [example](#)
 - [validator](#)
 - [post-function](#)
 - [work-log](#)
- [Using project properties to calculate custom sequence numbers](#)
 - [example](#)
 - [post-function](#)
 - [calculated-field](#)
 - [project-properties](#)
- [Set a date based on current date](#)
 - [example](#)
 - [post-function](#)
- [Setting the priority depending on the multiplication of custom fields](#)
 - [example](#)
 - [calculated-field](#)
 - [post-function](#)
- [Parse Email addresses to watchers list](#)
 - [example](#)
 - [post-function](#)
- [Set the assignee based on a condition](#)
 - [example](#)
 - [post-function](#)
- [Create a dynamic set of sub-tasks based on checkbox selection with unique summaries](#)
 - [example](#)
 - [post-function](#)
 - [custom-field](#)
 - [sub-task](#)
- [Create a static set of sub-tasks with unique summaries](#)
 - [example](#)
 - [post-function](#)
- [Triage Jira Service Desk email requests \(Move issues\)](#)
 - [example](#)
 - [post-function](#)
 - [move](#)
 - [transition-issue](#)
- [Moving story to "In Progress" when one of its sub-tasks is moved to "In Progress" \(Transition issues\)](#)
 - [example](#)
 - [post-function](#)
 - [transition](#)
- [Transition sub-tasks when parent is transitioned](#)
 - [example](#)
 - [post-function](#)
 - [sub-task](#)

- transition
 - outdated
- Transition only a sub-task among several ones
 - example
 - post-function
 - sub-task
 - transition
 - outdated
- Moving sub-tasks to "Open" status when parent issue moves to "In Progress"
 - example
 - post-function
 - sub-task
 - transition
 - outdated
- Moving story to "Ready for QA" once all its sub-tasks are in "Ready for QA" status
 - example
 - post-function
 - sub-task
 - transition
 - outdated