

Assign issue to current user if assignee is empty

On this page

- [Features used to implement the example](#)
- [Example: Assign issue to current user if assignee is empty](#)
- [Alternative implementation](#)
- [Other examples of that function](#)
- [Related Usage Examples](#)

Features used to implement the example

- [Set a field as a function of other fields](#)

Example: Assign issue to current user if assignee is empty

I would like to automatically **assign issue to current user** if **assignee is empty**, but if assignee is not empty, then nothing has to be done.

To implement this behavior we use post-function [Set a field as a function of other fields](#). We have two options, since this post-function have two kind of setting rules: **type 1**, based in regular expressions, and **type 2**, which uses boolean expressions. I will explain both solutions:

Using **type 1** setting rules, we use post-function [Set a field as a function of other fields](#) with the following configuration:

Field to be checked for matching with type 1 setting rules:	Assignee This field is only used by rules where conditional part is a regular expression written in brackets: <code>'(regular_expression)'</code>value
Target field to be set:	Assignee ~ [User] Field to be set by first matched setting rule. Type of the field is shown in square brackets. Check documentation on Virtual Fields to get information about suitable values for setting selected target field.
Setting rules: There are two types of setting rules, and both types can be combined in the same post-function. Rule formats: - type 1: <code>'(regular_expression)'</code> value - type 2: <code>'(boolean_expression)'</code> value Write only one rule per line. <i>value</i> may be a parsed text or a mathematical or time formula, depending on the type of selected <i>Target field</i> . Regular expression syntax	(.*)%{00020}

Note that:

- `^$` is regular expression for empty string, and is used to check whether **Assignee** is empty
- `%{00020}` is code for field **Current user**

There is a mistake in the screenshot of **Solution 1**: setting rule in the screenshot is `(.*)%{00020}` which is incorrect. Correct setting rule is: `(^$)%{00020}`

Once configured, transition "**Create issue**" will look like this:

Transition: Create Issue

Edit

View Properties

?

Create Issue

OPEN

This is the **initial** transition in the workflow.

Screen: None - initial transition does not have a view.

Validators 2

Post Functions 3

The following will be processed after the transition occurs

Add post function

1. Creates the issue originally.

2. The field **Assignee** will be set according to the evaluation of **Assignee** against the following set of rules:
(^\$)#Current user#

3. Fire a **Issue Created** event that can be processed by the listeners.

I have used transition "Create issue" in the example, but you can implement this post-function in any transition of your workflow.

Alternative implementation

Using type 2 setting rules, we use post-function **Set a field as a function of other fields** with the following configuration

Field to be checked for matching with type 1 setting rules:

Summary

This field is only used by rules where conditional part is a regular expression written in brackets: `'(regular_expression)'value`

Target field to be set:

Assignee - [User]

Field to be set by first matched setting rule. Type of the field is shown in square brackets.
Check [documentation on Virtual Fields](#) to get information about suitable values for setting selected target field.

Setting rules:

There are two types of setting rules, and both types can be combined in the same post-function.

Rule formats:

- type 1: `'(regular_expression)'value`

- type 2: `'(boolean_expression)'value`

Write only one rule per line.

value may be a parsed text or a mathematical or time formula, depending on the type of selected **Target field**.

[Regular expression syntax](#)

[%{00003}=null]{00020}

Notice that:

- Value selected in parameter "Field to be checked for matching..." has no effect in this case, since we are only using a **type 2** setting rule
- `%{00003}` is code for field **Assignee**

- `%{00020}` is code for field **Current user**

Once configured, transition "**Create issue**" will look like this:

Transition: Create Issue

EditView Properties?

Create Issue

OPEN

This is the **initial** transition in the workflow.

Screen: None - initial transition does not have a view.

Validators 2

Post Functions 3

The following will be processed after the transition occurs

Add post function

- Creates the issue originally.
- The field **Assignee** will be set according to the evaluation of **Summary** against the following set of rules:
`[%{Assignee}=null]#Current user#`
- Fire a **Issue Created** event that can be processed by the listeners.

I have used transition "**Create issue**" in the example, but you can implement this post-function in any transition of your workflow.

Other examples of that function

Page: [Add watcher depending on security level](#)
 Page: [Add watchers based on issue type](#)
 Page: [Add watchers depending on the value of a custom field](#)
 Page: [Assign issue based on the value of a Cascading Select custom field](#)
 Page: [Assign issue to a specific user based on a specific custom field value](#)
 Page: [Assign issue to current user if assignee is empty](#)
 Page: [Assign issue to current user if the user is not member of a certain project role](#)
 Page: [Change assignee based on a custom field](#)
 Page: [Change parent's status depending on sub-task's summary](#)
 Page: [Changing issue priority depending on issue description](#)
 Page: [Compose dynamic text by inserting field values in a text template](#)
 Page: [Copy "Due date" into a date type custom field in a linked issue if it's greater than current issue's "Due date"](#)
 Page: [Limit the number of hours a user can log per day](#)
 Page: [Make parent issue progress through its workflow](#)
 Page: [Rise priority if due date is less than 3 weeks away](#)
 Page: [Set "Due date" depending on the value of other fields, in case it's uninitialized](#)
 Page: [Set "Due date" to a specific day of next week no matter of date of creation this week](#)
 Page: [Set "Due date" to current date at issue creation if not initialized](#)
 Page: [Set a custom field "Urgency" depending on a combined value of issue's priority and "Impact" custom field](#)
 Page: [Set a date based on current date](#)
 Page: [Set a field based on reporter's email](#)

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](#)

Page: [Set a watcher at ticket creation depending on custom field's value](#)
 Page: [Set assignee depending on issue type](#)
 Page: [Set security level based on groups and project roles the reporter or creator are in](#)
 Page: [Set security level depending on reporter or creator](#)
 Page: [Set the assignee based on a condition](#)
 Page: [Set the value of a field of type "User Picker" depending on other field's value](#)
 Page: [Set watchers depending on the value of a custom field](#)
 Page: [Setting a custom field \(User Picker\) based on the value of another custom field \(Text Field\)](#)
 Page: [Setting a field's default value depending on another field](#)
 Page: [Setting the priority depending on the multiplication of custom fields](#)
 Page: [Transition an issue automatically depending on the value of a field](#)
 Page: [Unassign an issue when assigned to project leader](#)
 Page: [Update checkboxes custom field if a file has been attached during a transition](#)
 Page: [Using project properties to calculate custom sequence numbers](#)

- [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](#)