

# Log absence time on another issue

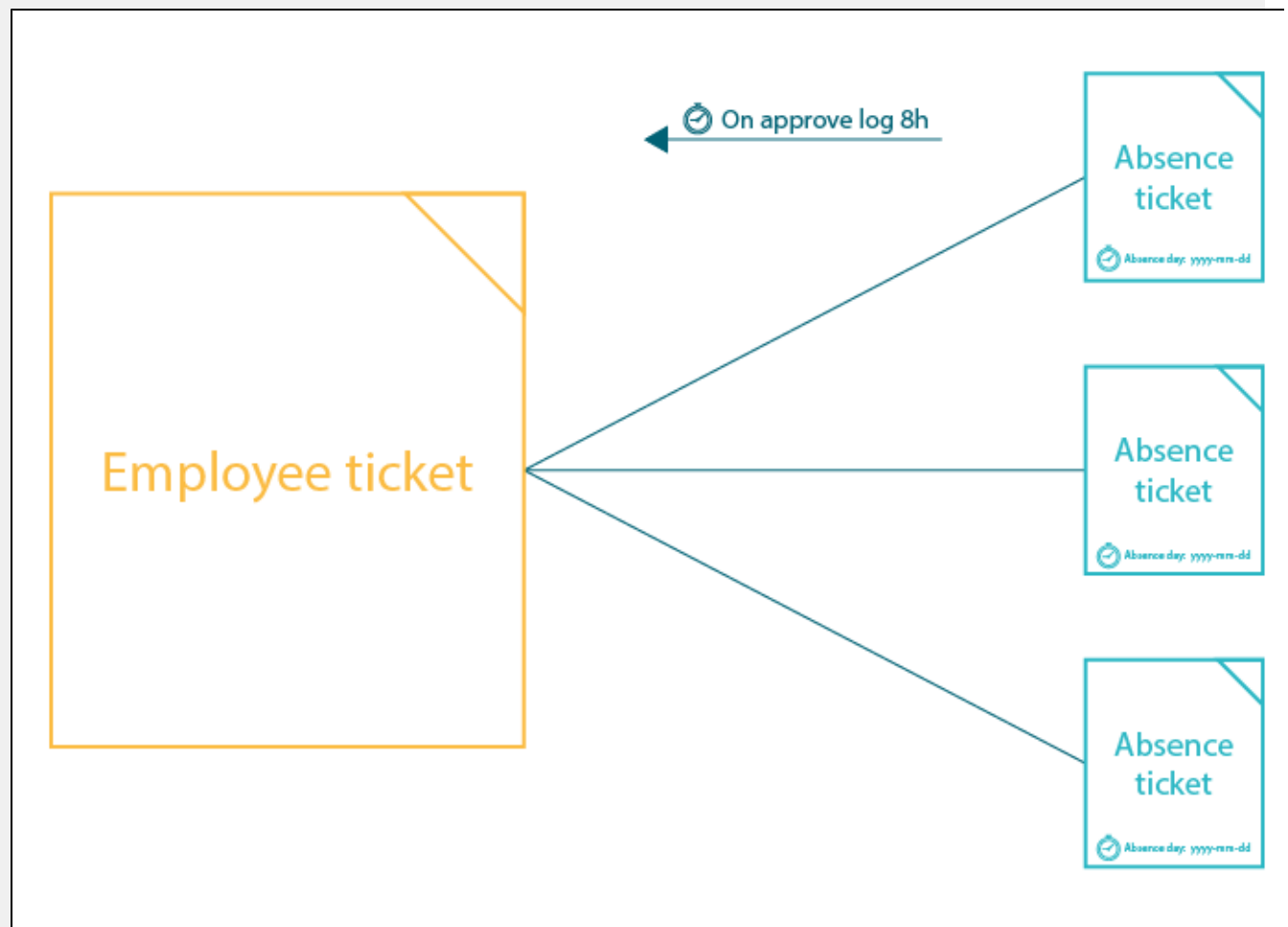
## On this page

[Prerequisites](#) | [Configuration](#) | [Screenshots](#) | [Related use cases](#)

## Use case

In this example, the [Log work](#) post function will be used to log work on another ticket.

In this case there is a Jira **ticket for each employee**. In addition, "**Absence**" tickets can be created for a **specific day**. If this ticket gets **approved**, **8 hours will be logged on the related employee ticket**.



The employee ticket must have the summary "**Employee <username>**".



## Prerequisites

First, we have to create a date custom field to enter the absence day.

## Set up the custom field

1

Create a Date Picker custom field

**Name:** Absence day

2

Add the created custom field to the screen(s) of the absence ticket.



## Configuration

1

Add the [Log work](#) post function to the desired transition or create a new transition.

2

### Target issue

Choose **Set target issue manually**

3

### Expression

Enter the following expression with mode set to **JQL**:

```
summary ~ "Employee %{issue.reporter}"
```

4

### Time spent

Enter the following number:

```
480
```

Since the time spent must be entered in minutes and 8h correspond to 480 minutes, this number has to be entered.

5

### Starting date-time

Enter the following expression:

```
stringToDate(%{issue.cfnnnnn} + " 08:00", "yyyy-MM-dd hh:mm")
```

**cfnnnnn** is the field code for the **Absence day** custom field we've previously created.

**Update the field code to the corresponding field code on your instance!**

6

## Work description

Enter a description like the following:

Absence

Feel free to change the description according to your needs!

## Additional options

7

### Remaining estimate

Choose **Auto adjustment**

8

### Conditional execution

Finally, we add a conditional execution so that the post function will only be executed if the issue's summary is **"Absence"**.

```
%{issue.summary} ~ "Absence"
```



## Screenshots

[Update parameters of the Log work Function for this transition.](#)

Update parameters of the Log work Function for this transition.

### Basic configuration \*

Define the required parameters needed to execute this post function.

Target issue \*

Set target issues manually (parser expression)

Select the issues to log work on.

Expression \*

```
1 summary ~ "Employee %{issue.reporter}"
```

Enter a JQL query and optionally use **field codes**, e.g. `%{issue.summary}`, to insert field values.

Time spent \*

```
1 480
```

The expression must return the time to be logged in minutes.

Starting date-time

```
1 stringToDate(%{issue.cf13201} + " 08:00", "yyyy-MM-dd hh:mm")
```

The expression must return the starting date-time of the work log entry. In case no expression is entered, current date-time is used.

Work description

```
1 Absence
```

Enter plain text and optionally use **field codes**, e.g. `%{issue.summary}`, to insert field values.

### Additional options

Remaining estimate

Auto adjustment

### Conditional execution

The post function will only be executed if the following condition (a **logical expression**) is met.

Expression

```
1 %{issue.summary} ~ "Absence"
```

The expression must return **true** or **false**.

> Run as

### Log work

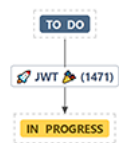
Documentation



Log work on one or multiple issues




### Workflow

JWT 3.0.0 Use cases





## Related use cases

Use case	JWT feature	Workflow function	Parser functions	Label
Add percentaged profit margin when closing issue		Log work		
Automatically log work spent in a specific status		Log work	timeDifference()	
Log absence time on another issue		Log work	stringToDate()	

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